# BEYOND IN-TASK ACCURACY: EVALUATING TASK-SPECIALIZED MEMORY UNDER RELEVANCE SHIFTS

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

## ABSTRACT

High accuracy on anticipated queries does not establish whether an LLM memory remains useful when different aspects of past experience become relevant. We isolate this evaluation gap by changing a memory writer's announced task while holding the history and decision rule fixed. In a controlled feature–reward task with exact scoring and reward-reversed histories, the writer is either promised that one feature dimension will matter or told that any dimension may matter. We freeze its memory and evaluate both announced and shifted queries. In the small task, cued memories from GPT-5.5 and GPT-5.4 achieve 100% in-task accuracy but 50.0–51.7% after a relevance shift. Uncued memories achieve 100% on both without exhausting the same 100-word allowance. Thus, identical in-task scores conceal different cross-query capabilities even when a broader useful record fits. The shift deliberately violates the cued writer's promise: specialization is appropriate for the announced task, not evidence of irrational forgetting. A separate 100-dimension task exposes model-dependent encoding and reading limitations rather than a uniform shift effect. A post-hoc retrieval audit distinguishes another source of error: all 302 non-tie lexical actions follow the selected reward sum, including 44 wrong full-history answers. Together, these results support evaluating anticipated-task accuracy, relevance-shift performance, and evidence use separately, rather than treating success on the anticipated task as a sufficient measure of memory quality.

## 1 INTRODUCTION

LLM agents use stored summaries and event records to reuse past experience (Park et al., 2023; Packer et al., 2023). What those records need to preserve depends on the questions they will support. For example, an agent recording outcomes by product category may later be asked about color. A note sufficient for the first question need not support the second. What does success on the anticipated task establish about a memory's usefulness after the task changes?

We distinguish three objects: a *history* of events, a *memory* written from that history, and a later *query* answered using the memory. *In-task accuracy* measures answers to the type of query announced when the memory was written. A *relevance shift* changes which feature of the same events determines the answer, while leaving the decision rule unchanged. In our task, each event has a reward and features such as category and color. The rule is to accept an offer if the rewards of matching events sum to a positive value. Figure 1 gives a worked example.

Prior work establishes the risks of compression before future queries and separates retrieval from reading (Zhu et al., 2026; Yang et al., 2026; Wu et al., 2025; Yu et al., 2026). We isolate a specific question within that problem: can changing only the writer's announced task produce memories with identical in-task scores but different performance after relevance shifts? Adapting the feature–reward task of Nicholas & Mattar (2026), we compare a writer promised one relevant feature with a writer told that any feature may matter. We freeze each memory before testing it. Reversing rewards in matched histories rules out success from a fixed action preference.

The shifted query breaks the first writer's promise. Specialization is sensible under that promise; we test the resulting memory's usefulness beyond it. We also ask whether a broader record fits within the same allowance, examine representation errors in a larger task, and audit whether retrieval errors follow the supplied evidence. Exact reward sums make these comparisons auditable.



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.

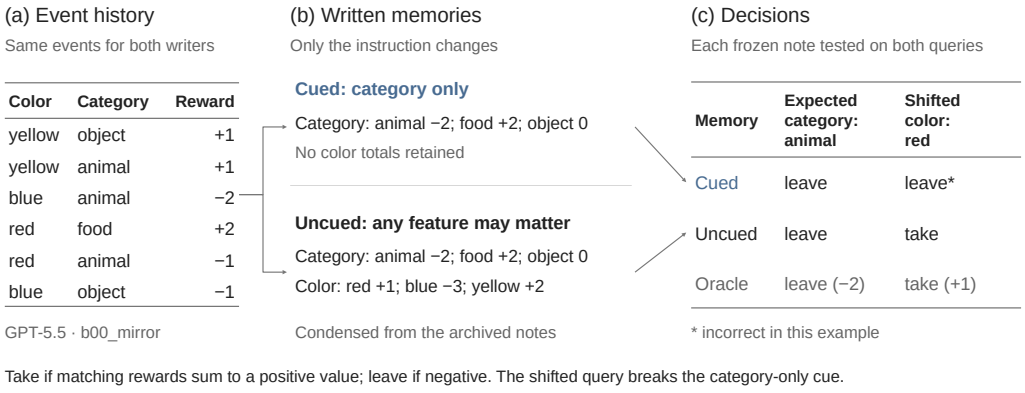


Figure 1: (a) Events from `b00_mirror`. (b) Condensed GPT-5.5 notes from cued and uncued writers. (c) Observed actions from testing each frozen note on both queries in separate calls. The asterisk marks an error against the full-history oracle. The color query violates the category-only cue.

Three findings organize the paper:

1. **Identical in-task accuracy can conceal different query coverage.** In the small task, cued memories score 100% on promised queries but 50.0–51.7% after a shift; uncued memories score 100% on both. Neither condition reaches the same 100-word ceiling. A broader useful memory fits, but the narrow writing instruction need not produce it.
2. **A failed summary need not imply an impossible memory budget.** In the separate 100-dimension task, uncued shifted accuracy is 57.5% for GPT-5.5 and 85.0% for GPT-5.4, although a 100-character code can store every exact answer given a shared decoder. This separates observed representation failures from capacity necessity.
3. **Missing events can change the answer a reader follows.** All 302 non-tie lexical actions agree with the selected event sum. The 53 full-history errors divide into 44 sign reversals and nine zero-sum cases. These repeated traces over 20 base cases provide a separate evidence-use diagnostic, not an explanation of every summary error.

These findings support separate checks of anticipated-task performance, performance after relevance changes, and use of the evidence actually supplied.

## 2 Related Work

**Memory architectures and evaluation.** Generative Agents retain natural-language experiences and retrieve them alongside reflections (Park et al., 2023). MemGPT manages different memory tiers (Packer et al., 2023), while A-MEM constructs linked, structured notes (Xu et al., 2025). LoCoMo evaluates extended conversational histories (Maharana et al., 2024); LongMemEval distinguishes indexing, retrieval, and reading, and tests extraction, temporal reasoning, updates, and abstention (Wu et al., 2025). These systems already establish the value of external memory and structured retrieval. LongMemEval also finds that complete recall need not ensure successful reading. We hold a decision rule fixed while varying the writer's announced task, then use selected-event sums to audit evidence use in a separate retrieval comparison.

**Compression before future queries.** TierMem identifies omissions caused by writing summaries before a later query and combines summary access with escalation to raw evidence (Zhu et al., 2026). ProMem examines proactive extraction and checks for missing information (Yang et al., 2026). DeferMem and LazyMem retrieve broad candidate pools before constructing query-conditioned evidence (Yin & Tang, 2026; Yu et al., 2026). MemRefine studies factual-content-aware compression of an existing memory store under explicit storage budgets (Kim et al., 2026). TierMem and LazyMem also analyze pipeline errors; we do not introduce the distinction between missing evidence and faulty reading. Our contribution is to manipulate the writer's announced task and use a decision

problem where both the complete-history answer and the answer implied by retrieved events are exactly computable. This complements naturalistic memory benchmarks with a controlled test of specialization and evidence use. We do not evaluate or claim to outperform these architectures.

**Context utilization and human memory.** Prompt compression trades input length against downstream performance (Jiang et al., 2023); long-context availability alone does not guarantee reliable use of evidence (Liu et al., 2024). We distinguish unavailable information from difficulty using an available cache through explicit controls. Nicholas & Mattar (2026) study how people use detailed past events when future relevance is uncertain. Their task logic motivates our benchmark, but our textual, stateless calls are not a replication of human cognition. Indeed, the generated memories frequently contain precomputed feature sums rather than item-level episodes.

## 3 METHODS

Table 1 maps the three evaluation stages and their memory controls. The small and larger tasks compare writing instructions; the retrieval study reuses the larger-task histories to test event selection and evidence use. All stages use the same reward-summing rule.

### 3.1 DECISION TASK AND PAIRED HISTORIES

Each event has categorical features and a reward. An offer specifies one feature value, such as `color=red`; the agent takes the offer if the matching rewards sum to a positive value and leaves it otherwise. In Figure 1, the two red events have rewards $+2$ and $-1$, so the correct action is to take the offer. Formally, let a history be $H = \{(x_i, r_i)\}_{i=1}^{n}$, where $x_i$ contains categorical features and $r_i$ is an integer reward. A query $q = (d, v)$ asks about dimension $d$ and value $v$. The offer value and correct action are

$$V(H, q) = \sum_{i=1}^{n} \mathbf{1}[x_{i,d} = v] r_i, \qquad a^*(H, q) = \mathbf{1}[V(H, q) > 0]. \tag{1}$$

Queries with zero full-history value are excluded. Both the writer and reader receive the complete rule. The reader returns a Boolean action and an estimated integer value; an exact programmatic oracle scores the action, without an LLM judge.

Each independently generated base history has a mirrored version $H^- = \{(x_i, -r_i)\}_{i=1}^{n}$. The features, identifiers, order, queries, contract, and action set remain unchanged, while $V(H^-, q) = -V(H, q)$. Consequently, correct actions must reverse. Inference treats both versions as one cluster; neither mirrored histories nor multiple queries are independent replications of the underlying generated case.

### 3.2 WRITING, FREEZING, AND TESTING MEMORY

**Write.** The writer receives the history, decision rule, and memory allowance, but no later query. The only experimental change is its relevance instruction: *cued* writers are promised that one named dimension will matter; *uncued* writers are told that any dimension may matter. **Freeze.** We save the resulting memory without revising it after a query appears. **Test.** A fresh call to the same model receives only the rule, saved memory, and current offer. Separate calls test an *expected* query about the announced dimension and a *shifted* query about another dimension. Both writing conditions use the same test queries. For uncued memories, "expected" is a matched evaluation label, not information disclosed to the writer. In-task accuracy refers to this expected-query score.

The shift contradicts the cued writer's promise. We compare categorical versus unspecified relevance, not calibrated responses to numerical query priors. Model parameters remain fixed throughout.

### 3.3 TASK SETTINGS AND MEMORY CONTROLS

An *exact cache* contains programmatically computed feature–reward sums; exact event selection instead supplies all query-matching events. These controls test whether adequate evidence supports correct decisions, not matched-storage efficiency. The small summary limit is prompted; the larger

Table 1: Study overview: three stages, two sets of histories. Retrieval reuses the larger-task histories, not a third independent sample. Summary allowances do not constrain raw histories, exact caches, or retrieved records.

| Stage and question | Histories and allowance | Memory conditions and controls |
|---|---|---|
| **Small task**<br>Does the writing cue change query coverage? | 30 base pairs (60 histories); 6 events; 2 or 4 dimensions.<br>Summaries: $\leq$100 words. | Cued and uncued summaries; full raw history; exact cache for the announced dimension; full exact cache. |
| **Larger task**<br>Do failures reflect encoding or use of available evidence? | 20 new base pairs (40 histories); 12 events; 100 binary dimensions.<br>Summaries: $\leq$500 characters. | Cued and uncued summaries; full raw history; full exact cache; exact matching-event selection (requested field and reward only). |
| **Retrieval + audit**<br>Does selection change the answer implied by the evidence? | Same 40 larger-task histories.<br>Return $k$ of 12 events: $k = 6$ primarily; $k = 3, 9$ in GPT-5.5. | Word BM25, field-pair BM25, and dense retrieval, compared with the existing uncued summary. Full-history scoring for all; selected-evidence audit for lexical retrievers only. |

limit is schema-enforced. All saved summaries obey their limit. Event-generation details are in Appendix A.

### 3.4 Retrieval study: selecting events after the query

This study retains raw events and retrieves a subset after the query arrives. Each retriever sees rendered event records and the query string, such as `f042=a`, and returns $k$ of twelve records. It receives no list of relevant events or oracle action, but the query exposes the field and value; field-pair indexing explicitly uses this schema.

Word BM25 splits records into alphanumeric tokens; field-pair BM25 keeps each field and value together. Dense retrieval ranks full-event embeddings by cosine similarity. Selected records retain all fields. We measure precision and recall against exact field equality and subsequent action accuracy. Implementation details are in Appendix D.

### 3.5 Scoring answers against full and retrieved evidence

The retrieval audit asks two questions: is the answer correct about the full history, and does it follow the rewards in the events the reader actually saw? For the lexical rankers, the generator and ranking code recover exactly which events were selected. If $S$ is that set, its query value is

$$V_S(H, q) = \sum_{i \in S} \mathbf{1}[x_{i,d} = v] r_i, \qquad a_S(H, q) = \mathbf{1}[V_S(H, q) > 0] \quad \text{if } V_S(H, q) \neq 0. \tag{2}$$

We score the action against two references: $a^*$ from the full history and $a_S$ from the selected evidence. An action can agree with $a_S$ while being wrong against $a^*$. This distinguishes disagreement with the delivered evidence from a change in the answer implied by selection. Zero selected sums have no unique prescribed action and are reported separately.

This reference measures agreement with the visible event sum, not an optimal policy under missing information. Abstaining or requesting evidence is not available in this binary task. Agreement does not establish internal arithmetic. Dense selections and retrieval-time numeric estimates were not saved, limiting the audit to lexical actions.

### 3.6 Models, statistical tests, and analysis timing

The archive records `gpt-5.5-2026-04-23` for the writing studies, `gpt-5.5` for the retrieval follow-up, and `gpt-5.4` for replication. All use low reasoning effort and structured outputs. Model names are the requested identifiers stored by the runners; aliases need not be immutable snapshots. The decision task has one response per frozen memory, condition, and query. The second

Table 2: Small-task accuracy (%). Each cell has 60 decisions from 30 independent mirrored pairs. "Exp." denotes the matched expected query; "shift" denotes the other dimension. Uncued writers see neither query in advance.

| Memory | 5.5 exp. | 5.5 shift | 5.4 exp. | 5.4 shift |
|---|---|---|---|---|
| Cued summary | 100.0 | 51.7 | 100.0 | 50.0 |
| Uncued summary | 100.0 | 100.0 | 100.0 | 100.0 |
| Full event history | 100.0 | 100.0 | 100.0 | 100.0 |
| Cued exact cache | 100.0 | 50.0 | 100.0 | 50.0 |
| Full exact cache | 100.0 | 100.0 | 100.0 | 100.0 |

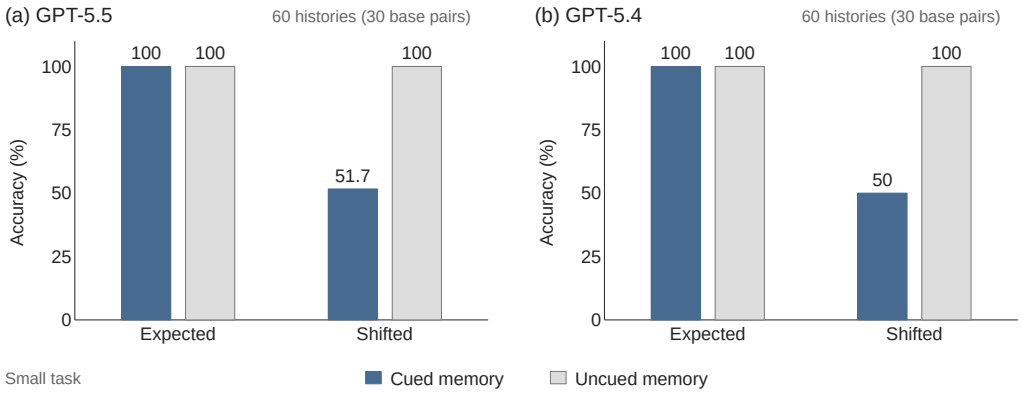


Figure 2: Small-task accuracy for (a) GPT-5.5 and (b) GPT-5.4. Both conditions are perfect on expected queries; only uncued memories remain perfect after the shift. Bars summarize 60 decisions (30 mirrored pairs). Raw-event controls are 100%; paired intervals are in the text.

model repeats the same synthetic cases, providing a model comparison rather than independent task replication.

We report effects averaged within base pairs, two-sided exact sign-flip tests, and percentile bootstrap intervals over base pairs. For the three retrieval comparisons to each model's uncued summary, the threshold is $0.05/3$. Appendix B retains the registered retrieval sign tests over 40 histories alongside the manuscript's pair-clustered sensitivity tests, preregistered gates, and deviations.

The memory-length audit, cue–query interaction, and deterministic retrieval reconstruction were added after inspecting the results. These are post-hoc analyses of the same frozen responses, not additional independent experiments. All original outcomes, including failed registered gates, remain reported.

## 4 Results: Memory Writing Under Relevance Shifts

We first compare the two writing instructions in the small task, then examine the larger task's representation errors. A compact-code control and a post-hoc interaction test clarify what these comparisons establish.

### 4.1 Identical In-Task Scores, Different Relevance-Shift Outcomes

Table 2 and Figure 2 show the small-task results. Cued summaries answer expected queries perfectly in both models, but shifted-query accuracy is 51.7% for GPT-5.5 and 50.0% for GPT-5.4. Uncued summaries achieve 100% on both query types, as do raw events and full exact caches. For GPT-5.5, uncued minus cued shifted-query accuracy is $+48.3$ percentage points (pp), 95% CI $[45.0, 50.0]$, $p = 3.73 \times 10^{-9}$ over 30 pairs. GPT-5.4 reproduces the contrast: $+50.0$ pp, CI $[50.0, 50.0]$, $p = 1.86 \times 10^{-9}$.

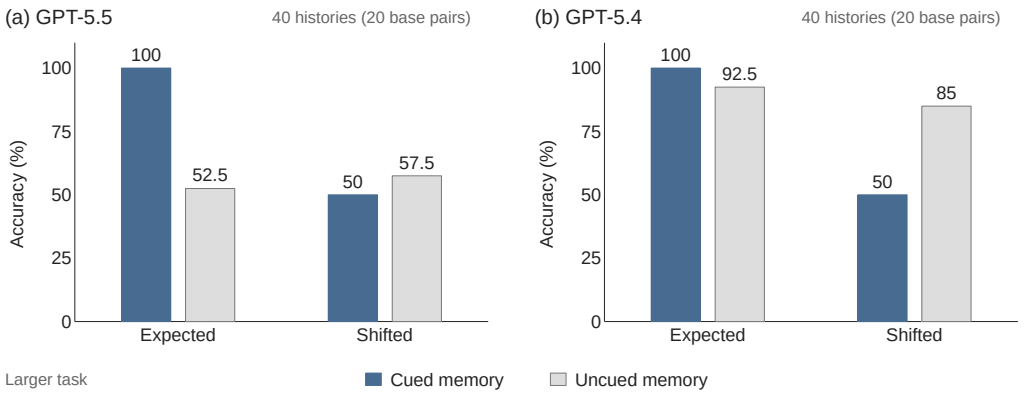


Figure 3: Larger-task accuracy for (a) GPT-5.5 and (b) GPT-5.4. Bars summarize 40 decisions (20 mirrored pairs); raw-event controls are 100%. Cued memories specialize successfully, while uncued performance varies. Comparison with Figure 2 changes several task factors together and does not isolate dimensionality.

The stored text illustrates the contrast: on `b00_mirror`, GPT-5.5's cued memory lists category sums, while its uncued memory also lists color sums. These aggregates suffice for the decision rule. Neither model's natural memories contain original item identifiers. The visible sums support semantic caching as a description of the output, but identifier absence alone cannot identify an internal cognitive process.

The cued memories fail the registered flexible-backup criterion, while uncued memories support both queries. Neither condition reaches the word ceiling: the largest observed memories use 76 words in GPT-5.5 and 64 in GPT-5.4. Median cued/uncued lengths are 45/43 and 38/38.5 words, respectively (Appendix B.1). Together with perfect uncued accuracy, this shows that a broader useful representation fits within the same interface. The experiment measures which information the writer selects; it does not demonstrate a necessary accuracy–storage trade-off or calibrated allocation under a probabilistic relevance prior.

### 4.2 Separate encoding and reading limitations in the larger task

At 100 dimensions, GPT-5.5's uncued shifted-query accuracy is 57.5%, while its raw-event, exact-selection, and full-cache controls remain perfect (Table 3; Figure 3). Exact selection improves on its uncued summary by $42.5$ pp, CI $[35.0, 50.0]$, $p = 1.53 \times 10^{-5}$ over 20 pairs. Its cued expected accuracy remains 100%, confirming that the 500-character interface can retain sufficient information for an announced dimension. Its uncued expected accuracy is only 52.5%: this condition has a broader encoding/use problem, not just a loss following query shift.

GPT-5.4 is a substantive counterexample to a uniform compression-failure claim. It often uses compact ranges of feature indices to encode the signs of many aggregates, attaining 85.0% shifted-query accuracy. Its exact-selection gain is $15.0$ pp, CI $[5.0, 25.0]$, $p = 0.03125$. This misses the registered 20-point effect threshold. Moreover, its larger explicit full-cache control achieves only 52.5% shifted accuracy, despite perfect raw-event and exact-selection performance. The preregistered larger-task pattern therefore does not fully replicate in GPT-5.4. Available information can also be hard for a reader to use in a particular representation.

### 4.3 A compact construction bounds the capacity interpretation

Each larger-task history has total reward zero, so its binary feature sums satisfy $V(H, (d, b)) = -V(H, (d, a))$. Encoding one sum per dimension with a shared 19-symbol alphabet preserves every exact answer in 100 characters; a three-symbol alphabet suffices for actions alone. Appendix C gives the construction. The shared decoder is assumed, not learned or tested.

Table 3: Larger-task accuracy (%). Each cell has 40 decisions from 20 pairs. Only natural summaries have the 500-character limit. Exact event selection also removes irrelevant fields, whereas full histories retain them.

| Memory | 5.5 exp. | 5.5 shift | 5.4 exp. | 5.4 shift |
|---|---|---|---|---|
| Cued summary | 100.0 | 50.0 | 100.0 | 50.0 |
| Uncued summary | 52.5 | 57.5 | 92.5 | 85.0 |
| Full event history | 100.0 | 100.0 | 100.0 | 100.0 |
| Exact event selection | 100.0 | 100.0 | 100.0 | 100.0 |
| Full exact cache | 100.0 | 100.0 | 75.0 | 52.5 |

Table 4: The same action evaluated against two references. All 302 non-tie lexical actions agree with the selected evidence, including 44 full-history errors. Counts pool shifted queries across settings that reuse 20 base pairs; they are descriptive traces, not independent trials.

| Selected vs. full sum | Decisions | Full-history correct | Selected agreement |
|---|---|---|---|
| Same nonzero sign | 258 | 258/258 | 258/258 |
| Opposite nonzero sign | 44 | 0/44 | 44/44 |
| Zero selected sum | 18 | 9/18 | Undefined |

This establishes feasibility, not efficient construction or reliable model decoding. Uncued medians are 286 and 263 characters; only one of 80 memories reaches 500. The experiment tests chosen encodings and their use, not an information-theoretic limit. Event count, vocabulary, and budget units change together, so the task comparison does not isolate a scaling variable.

### 4.4 The announcement effect at both task settings

To distinguish task-specific specialization from a general change in accuracy, we compute the post-hoc interaction

$$I = (\mathrm{Acc}_{\mathrm{cued,exp}} - \mathrm{Acc}_{\mathrm{uncued,exp}}) - (\mathrm{Acc}_{\mathrm{cued,shift}} - \mathrm{Acc}_{\mathrm{uncued,shift}}). \tag{3}$$

Positive $I$ means the cue's accuracy advantage is larger for promised than shifted queries. The effects are 48.3/50.0 pp in the small task and 55.0/42.5 pp in the larger task for GPT-5.5/GPT-5.4, respectively. Pair-clustered intervals appear in Appendix B.2. Specialization is observable in both settings, but only the small task shows perfect performance from an uncued memory. The interaction does not turn the larger task's general encoding failures into query-shift failures.

## 5 Results: retrieval and evidence use

Unlike the writing intervention, this follow-up retains raw events and varies retrieval. Exact event sums distinguish action-level disagreement with supplied evidence from an answer changed by selection; they do not decode discarded summary content.

### 5.1 An exact account of the observed lexical errors

The reconstructed lexical selections reproduce archived precision, recall, and full-history answers for all 640 records. This post-hoc audit compares existing shifted-query actions with both references in Equations 1 and 2, without new model calls.

Of 320 shifted-query lexical decisions, 302 have a nonzero selected sum. Every one follows its sign (Table 4). For 258, the selected and full-history sums imply the same action. For 44, omitted relevant events reverse the full-history answer, and the model follows the selected evidence's opposite sign. The remaining 18 have zero selected sum; nine happen to match the nonzero full-history answer.

Thus the 53 lexical errors divide into 44 answer reversals and nine zero-sum cases, with no observed non-tie deviation from the available-evidence rule. The audit localizes the observable mismatch: the

Table 5: GPT-5.5 retrieval on shifted queries, $k = 6$. Accuracy is a percentage; precision and recall are fractions. Tests compare each retriever with the uncued summary and cluster the 40 histories into 20 base pairs.

| Retriever | Accuracy | Precision | Recall | $p$ (pairs) |
|---|---|---|---|---|
| Field-pair BM25 | 97.5 | 0.825 | 0.954 | $3.05 \times 10^{-5}$ |
| Word BM25 | 75.0 | 0.442 | 0.503 | 0.143 |
| Dense embedding | 70.0 | 0.483 | 0.558 | 0.386 |

model's action follows the selected events, while those events imply a different answer from the full history. It does not establish whether the model recognized missing evidence or performed exact internal arithmetic. Per-model, index, and depth counts appear in Appendix D.1.

### 5.2 Which selections produce the mismatch?

At $k = 6$, field-pair BM25 reaches 97.5% shifted-query accuracy in both models. Word BM25 reaches 75.0% and dense retrieval 70.0% (Table 5). Against GPT-5.5's 57.5% uncued summary, the field-pair gain is 40.0 pp, pair-bootstrap CI $[30.0, 47.5]$, $p = 3.05 \times 10^{-5}$. The other gains are 17.5 and 12.5 pp, with pair-level $p = 0.143$ and 0.386; neither meets the corrected threshold. Their non-significance does not establish zero benefit. Against GPT-5.4's 85.0% uncued summary, the field-pair gain is 12.5 pp ($p = 0.125$), so significant improvement over a natural summary is not a cross-model result.

The lexical difference follows from the event format. Every record contains each dimension name and many occurrences of both values. Splitting `f042=a` loses the field–value binding; the whole pair occurs exactly in matching records. At $k = 6$, mean shifted recall is 0.954 for field pairs, 0.503 for words, and 0.558 for dense retrieval. Pair BM25 therefore acts much like an exact structured filter, the natural ceiling for these equality queries. This is a binding-preservation control, not a new ranking algorithm. Appendix D details the lexical score; the experiment does not identify the dense retriever's internal failure mechanism.

### 5.3 Retrieval depth and complete evidence

The GPT-5.5 depth comparison increases field-pair accuracy from 85.0% at $k = 3$ to 97.5% at $k = 6$ and 100% at $k = 9$ (Figure 4). Recall rises from 0.612 to 0.954 to 1.000 while precision falls from 1.000 to 0.825 to 0.589. Thus, rejecting every irrelevant record is unnecessary for correct decisions in this task. At $k = 9$, word and dense retrieval remain at 77.5% and 85.0% accuracy, respectively. The $k = 9$ field-pair prediction was recorded after inspecting $k = 6$ and offline retrieval statistics, before collecting its model decisions; it is a targeted follow-up on the same cases.

Across both models and all recorded retrieval depths, all 136 shifted-query decisions with complete recall are correct; 241 of 344 incomplete-recall decisions are also correct. Complete recall is sufficient for the observed decisions, but not necessary. The exact lexical audit explains why recall alone cannot diagnose an individual error: missing an event matters for the action only if it changes the sign or leaves a tie. These counts reuse histories and do not add independent replications of the earlier audit.

## 6 Discussion

**A perfect score answers only the question that was tested.** In the small task, both writing instructions produce memories that answer expected questions perfectly. Asking about a different feature separates them: the cued memories fall to 50.0–51.7%, while uncued memories remain at 100%. The cued writer followed its instruction. The lesson is therefore not to avoid specialization, but to distinguish a memory suited to a known task from one that also supports other questions about the same history.

**What this changes in a memory evaluation.** An in-task test would rank these memories equally. Adding shifted queries reveals the difference without changing the history, rule, or writing allowance.

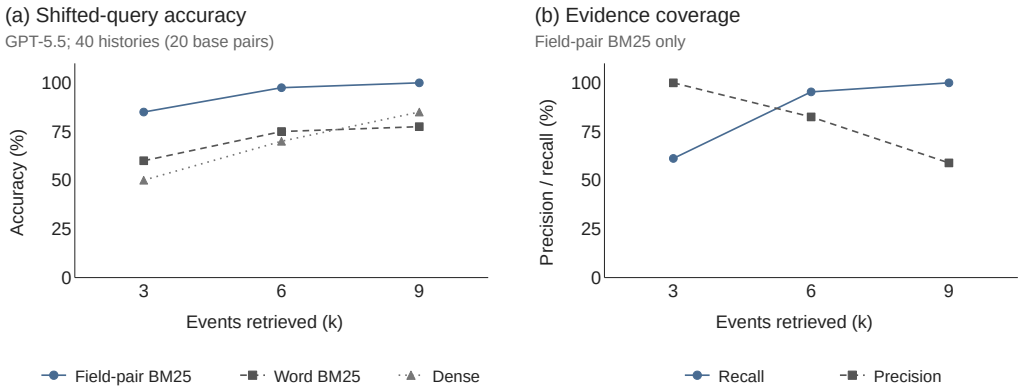


Figure 4: Retrieval on the same 40 GPT-5.5 histories. (a) Decision accuracy. (b) Field-pair recall (coverage of relevant events) and precision (relevant fraction returned). Lines connect measured depths; Appendix D reports budget statistics.

For systems intended to reuse experience across tasks, such tests can check which kinds of later questions remain answerable. A separate capacity check prevents a different mistake: interpreting a poor summary as proof that the storage limit made success impossible. Here, successful uncued memories and the compact-code construction provide those controls.

**A wrong answer need not be a failure to use the supplied evidence.** In 44 lexical cases, the retrieved events imply the opposite action from the full history, and the reader follows that incomplete evidence. Repeating the same computation on those events would not recover the full-history answer. The retrieval comparison shows a practical distinction: returning more relevant events can improve decisions even if more irrelevant events come with them. This motivates checking both the selected evidence and the reader's response, rather than treating all incorrect answers as the same error.

**Implications for agents that reuse experience.** In a long-running social simulation, a past interaction may matter for a different reason at a later time. Our results motivate testing stored memories against such changes before relying on their in-task scores. Application to EconSimulacra or other agent loops is future work, not a demonstrated transfer. The separate history-evidence control remains in Appendix E and is not evidence for the relevance-shift claim.

### 6.1 Limitations

The study has 30 small-task and 20 larger-task base pairs and two closely related proprietary models. It evaluates one-shot external memory formation, not repeated consolidation. Violating an explicit promise tests robustness to task mismatch, not irrationality or uncertainty calibration. The task settings change several factors together, and the larger-task pattern only partially replicates across models. Raw and retrieved records are not storage-matched to summaries; an exact selector additionally removes irrelevant fields. The compact construction assumes a shared decoder not tested with the models, and the field-pair index exploits literal equality queries. The selected-evidence audit is post hoc, excludes unrecoverable dense selections, and measures actions rather than internal arithmetic. Independent model families, probabilistic query priors, and repeated consolidation are needed to establish how far these patterns generalize.

## 7 Conclusion

Our controlled intervention produces identical 100% in-task scores but sharply different shifted-query outcomes, even when a broader useful memory fits within the same allowance. Representation controls and the retrieval audit separately test stored information, selected evidence, and its use. Together, these comparisons provide an auditable way to assess what an agent's memory supports beyond its announced task, rather than treating a wrong answer as a diagnosis of the failure.

## REPRODUCIBILITY STATEMENT

The task generators, frozen responses, preregistrations, and analysis scripts underlie every numerical table. Appendix A records models, seeds, prompts, sample counts, and archive limitations. Appendix B distinguishes registered tests from later sensitivity analyses. The manuscript's tables and figures are regenerated offline from the archived JSON responses; no additional model responses are used in preparing this draft.

## ETHICS STATEMENT

The reported experiments use synthetic textual histories and do not collect human participant data. They do not validate LLM agents as substitutes for human behavior. Claims about memory mechanisms are limited to observable external records and actions; no access to private model reasoning or human-like mental states is assumed.

## AI USE STATEMENT

Generative AI tools assisted with research development and literature review, code and analysis, and manuscript preparation, including figures. Experimental model use is documented in the methods and appendix. The authors are responsible for verifying the work and for the final manuscript.

## REFERENCES

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, 2023. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825/.

Minjae Kim, Jinheon Baek, Soyeong Jeong, and Sung Ju Hwang. MemRefine: LLM-guided compression for long-term agent memory. arXiv:2606.13177, 2026. URL https://arxiv.org/abs/2606.13177.

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/.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. arXiv:2402.17753, 2024. URL https://arxiv.org/abs/2402.17753.

Jonathan Nicholas and Marcelo G. Mattar. Episodic memory facilitates flexible decision-making via access to detailed events. *Nature Human Behaviour*, 10(4):760–776, 2026. doi: 10.1038/s41562-025-02383-3. URL https://www.nature.com/articles/s41562-025-02383-3.

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. 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*, 2023. URL https://arxiv.org/abs/2304.03442.

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://arxiv.org/abs/2410.10813.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.12110.

Chengyuan Yang, Zequn Sun, Wei Wei, and Wei Hu. Beyond static summarization: Proactive memory extraction for LLM agents. arXiv:2601.04463, 2026. URL https://arxiv.org/abs/2601.04463.

Jianing Yin and Tan Tang. DeferMem: Query-time evidence distillation via reinforcement learning for long-term memory QA. arXiv:2605.22411, 2026. URL https://arxiv.org/abs/2605.22411.

Jing Yu, Yibo Zhao, Jiaming Zhang, and Xiang Li. LazyMem: Retrieve broadly, construct selectively for efficient long-term agent memory. arXiv:2607.22690, 2026. URL https://arxiv.org/abs/2607.22690.

Qiming Zhu, Shunian Chen, Rui Yu, Zhehao Wu, and Benyou Wang. From lossy to verified: A provenance-aware tiered memory for agents. arXiv:2602.17913, 2026. URL https://arxiv.org/abs/2602.17913.

# A PROTOCOL AND ARCHIVED RESPONSES

All results are computed from archived model outputs. The small study uses seed 20260730 and 30 base pairs; the larger study uses seed 20260731 and 20 base pairs. Retrieval reuses the larger histories and their query assignments. There are 120 generated memories and 600 decisions per model in the small study, 80 memories and 400 decisions per model in the larger study, and 240 decisions per retrieval model/depth setting. No additional sampling is implied by plotting the same data in multiple figures.

**Event generation.** The small task has 15 base pairs with two four-valued dimensions and 15 with four binary dimensions. Each history has six events with shuffled rewards $(-2, -1, -1, 1, 1, 2)$. The larger task has twelve events with 100 binary dimensions and three copies of each reward in $\{-2, -1, 1, 2\}$. Each base history is paired with its reward-reversed version.

Table 6: Recorded model requests and counts per output file. "Writes" counts generated memory records; decision counts exclude those calls. All reasoning effort settings are low.

| Study | Model request | Writes | Decisions |
|---|---|---|---|
| Small | `gpt-5.5-2026-04-23` | 120 | 600 |
| Small | `gpt-5.4` | 120 | 600 |
| Larger | `gpt-5.5-2026-04-23` | 80 | 400 |
| Larger | `gpt-5.4` | 80 | 400 |
| Retrieval, each $k = 3, 6, 9$ | `gpt-5.5` | 0 | 240 |
| Retrieval, $k = 6$ | `gpt-5.4` | 0 | 240 |
| History, each order batch | `gpt-5.5-2026-04-23` | 128 | 1536 |
| History, each order batch | `gpt-5.4-2026-03-05` | 128 | 1536 |

**Prompt and sampling details.** Calls use the Chat Completions interface, a single user message, a strict JSON output schema, and no explicitly specified temperature. The model API remains stochastic; the generators' seeds determine cases, not model sampling. Writing and decision calls are separate, and the decision has no hidden access to the writer's transcript. The six-event writer's 100-word limit is a prompt instruction; its observed maximum is 76 words for GPT-5.5 and 64 for GPT-5.4. The larger writer is both prompted and constrained by a JSON string `maxLength` of 500. All stored larger-task memories satisfy that bound.

**Archive limitations.** The response JSON files preserve natural memories, actions, correctness, configuration, and most token totals. Retrieval output files preserve precision and recall, but not the ranked event identifiers, dense embedding vectors, or model-generated estimated values. The deterministic lexical rankings can be reconstructed from the generator and retriever code; the exact dense rankings cannot be recovered from these outputs alone. The initial GPT-5.5 $k = 6$ retrieval file has zero token counters due to an accounting bug, despite recording 240 decisions. Therefore, this paper does not report complete cost or latency comparisons. Requested model identifiers are retained, but backend response-version metadata are not sufficient to establish immutable alias behavior across runs.

## A.1 EXACT CONTRACT AND WRITING INSTRUCTIONS

The small-task contract is:

> Complete task contract: You previously observed six items. Each item had several features and an integer reward. A later offer names exactly one feature value. The offer's value is the sum of the rewards of every prior item possessing that feature value. Take an offer if and only if this sum is positive; leave it if the sum is negative. Each round is independent.

The cued instruction adds:

> Before seeing these episodes, you were told that every later offer would concern the [announced dimension] feature.

The uncued instruction replaces it with:

> You do not yet know which feature dimension later offers will concern. It may be any feature shown below.

Both append:

> Write a compact long-term memory of these experiences. This memory will be the only record available at decision time. Use at most 100 words. Do not assume an offer that has not been revealed.

The complete event list follows. The decision call repeats the contract, inserts the frozen memory and current query, and requests the exact estimated offer value supported by memory and whether to take it. It instructs the model to make its best decision without inventing episodes if memory is insufficient.

In the larger task, the contract specifies twelve items with binary features and repeats Equation 1. The cued writer is told, "You were promised that every later offer would concern only dimension [announced dimension]." The uncued writer is told, "You do not know which of the 100 dimensions will matter later. The relevant dimension will be revealed only after this memory is frozen." Both are asked to write the most useful memory within 500 characters. Exact strings and event formatting are in the archived runner files.

The decision schema is the same for summary, raw, cache, and retrieval conditions:

```
{"type": "object",
 "properties": {"take": {"type": "boolean"},
                "estimated_value": {"type": "integer"}},
 "required": ["take", "estimated_value"],
 "additionalProperties": false}
```

### A.2 Representation audit example

The following excerpts are from GPT-5.5 on `b00_mirror`. The cued memory contains:

> Offers will concern category only. Sum rewards by category from prior items: object = +1 + (-1) = 0; animal = +1 + (-2) + (-1) = -2; food = +2.

The uncued memory contains:

> Feature reward sums for future offers: color yellow = +2 (take), blue = -3 (leave), red = +1 (take). Category object = 0, animal = -2 (leave), food = +2 (take).

These examples illustrate aggregate selection; all accuracy results use the full case set. They are observable output text and are not private internal reasoning traces.

## B Registered plans and analysis changes

The archive contains timestamped study plans, model replication criteria, and a later retrieval-budget addendum. The small and larger task plans were recorded before their corresponding outcomes; the GPT-5.4 model plan was recorded after GPT-5.5 inspection and before GPT-5.4 collection. Retrieval baselines were specified after inspecting the exact-selection upper bound. The depth addendum used offline retrieval metrics available before the new decision calls. These stages share cases and are not wholly independent confirmations.

**Inferential unit.** Small- and larger-task contrasts use the mean of the two mirrored responses per base case. The exact test enumerates the sign-flip distribution of these cluster differences, computed by dynamic programming. A two-sided statistic counts absolute sums at least as large as observed. Percentile intervals use 100,000 bootstrap resamples of base pairs (seed 20260730 for the episodic analyses). A degenerate interval at 100% reflects uniformly successful observed pairs under this

Table 7: Selected registered gates. Results are descriptive pass/fail judgments against frozen study-specific criteria, not a joint test across the project.

| Criterion | GPT-5.5 | GPT-5.4 |
|---|---|---|
| Small: adequate full-cache and raw-event decisions | Pass | Pass |
| Small: uncued memory supports shifted queries | Pass | Pass |
| Small: cued memory retains a flexible backup | Fail | Fail |
| Larger: full-cache control above registered minimum | Pass | Fail |
| Larger: exact selection gains at least 20 pp with $p < 0.05$ | Pass | Fail |
| Larger model replication: uncued shift accuracy at most 75% | — | Fail |
| History: complete-contract effect at least 0.15 with $p < 0.05$ | Fail | Fail |

resampling scheme; it is not a population certainty statement or a repeated-sampling confidence bound for model accuracy.

The retrieval plan instead specified a sign test over the 40 histories. Since the mirrored variants share an underlying generated case, we add clustered tests over 20 pairs. These manuscript checks were run on already observed outcomes on September 15, 2026 and are sensitivity analyses, not new preregistered evidence. Against GPT-5.5's uncued summary, the registered history-level $p$ values for pair BM25, word BM25, and dense retrieval are $0.0000305$, $0.1435$, and $0.3593$. Their corresponding pair-clustered values are $0.0000305$, $0.1435$, and $0.3863$. Table 8 reports the clustered comparisons for both models. The corrected criterion is $p < 0.0167$ within each model's three retrieval-versus-summary comparisons.

Table 8: Sensitivity analyses against each model's own uncued larger-task summary. Differences and intervals are in percentage points.

| Model | Retriever | $\Delta$ (pp) | 95% CI (pp) | $p$ |
|---|---|---|---|---|
| 5.5 | Field-pair BM25 | +40.0 | [+30.0, +47.5] | $3.05 \times 10^{-5}$ |
| 5.5 | Word BM25 | +17.5 | [-2.5, +35.0] | 0.143 |
| 5.5 | Dense embedding | +12.5 | [-10.0, +32.5] | 0.386 |
| 5.4 | Field-pair BM25 | +12.5 | [+0.0, +25.0] | 0.125 |
| 5.4 | Word BM25 | -10.0 | [-30.0, +10.0] | 0.492 |
| 5.4 | Dense embedding | -15.0 | [-40.0, +7.5] | 0.335 |

**Separation from earlier exploratory work.** The project began with embodied-agent behavior, homeostatic drives, and obligation prompts. Those investigations motivated later controls but are not pooled into this paper's tests. The current results do not require a favorable outcome from those earlier experiments.

### B.1 Memory-length audit

The September 16, 2026 audit recomputes whitespace-delimited word counts for the small task and Unicode character counts for the larger task from every stored memory string. Table 9 reports both writing conditions; the units must not be compared across tasks. An unused allowance does not prove that the instruction exerted no pressure, but length alone cannot establish a binding storage trade-off. No semantic-coverage score is inferred from missing dimension names: compact codes can omit literal field labels while encoding their values.

### B.2 Cue–query interaction

Equation 3 is computed within each history and then averaged over its original/mirrored pair. Table 10 uses 100,000 base-pair bootstrap resamples with seed 20260916. These intervals and exact sign-flip tests are post-hoc checks of the same four datasets, not new confirmations. The analysis output also records Bonferroni-adjusted $p$ values for the family of four interaction tests. The comparison cannot measure calibration to a query prior: the original prompts used only a categorical promise or no announced dimension.

Table 9: Observed memory lengths. Small-task units are words (limit 100); larger-task units are characters (limit 500). $n$ counts memory records, including mirrored histories; "at limit" is the number exactly at the cap.

| Task | Model | Memory | $n$ | Median | Range | At limit |
|---|---|---|---|---|---|---|
| Small | 5.5 | Cued | 60 | 45 | 31–67 | 0 |
| Small | 5.5 | Uncued | 60 | 43 | 24–76 | 0 |
| Small | 5.4 | Cued | 60 | 38 | 18–51 | 0 |
| Small | 5.4 | Uncued | 60 | 38.5 | 26–64 | 0 |
| Larger | 5.5 | Cued | 40 | 146.5 | 90–228 | 0 |
| Larger | 5.5 | Uncued | 40 | 286 | 198–500 | 1 |
| Larger | 5.4 | Cued | 40 | 157.5 | 84–250 | 0 |
| Larger | 5.4 | Uncued | 40 | 263 | 145–461 | 0 |

Table 10: Post-hoc cue–query interaction. Positive values indicate that the cue's accuracy advantage is greater for promised than shifted queries. Intervals resample independent base pairs, not individual calls.

| Task | Model | Pairs | $I$ (pp) | 95% CI (pp) |
|---|---|---|---|---|
| Small | 5.5 | 30 | 48.3 | [45.0, 50.0] |
| Small | 5.4 | 30 | 50.0 | [50.0, 50.0] |
| Larger | 5.5 | 20 | 55.0 | [45.0, 65.0] |
| Larger | 5.4 | 20 | 42.5 | [30.0, 55.0] |

## C WHY 500 CHARACTERS IS NOT A PROVEN STORAGE BARRIER

For every larger-task history, the total positive reward is 9 and total negative reward is $-9$. For each binary dimension $d$, define $z_d = V(H, (d, a))$. Then $z_d \in \{-9, -8, \ldots, 9\}$ and $V(H, (d, b)) = -z_d$. A predetermined alphabet of 19 symbols can encode each $z_d$ with one symbol in a fixed dimension order. A 100-character string therefore encodes every exact query value in this specific task, given a shared decoder. A three-symbol sign code would suffice if only the Boolean action were required.

This construction is an offline observation about the task's sufficient statistics, not an additional LLM experiment. The runs did not supply or test this coding convention. It establishes that observed summary failure cannot be interpreted as an information-theoretic impossibility under the character budget. It also explains why a model that spontaneously chooses a better encoding can perform much better without a larger memory. Character count, token count, stored bits, and decoding difficulty must be distinguished.

## D ADDITIONAL RETRIEVAL DIAGNOSTICS

Both BM25 variants use $k_1 = 1.5$ and $b = 0.75$. The dense baseline embeds each full event with `text-embedding-3-small` and ranks by cosine similarity. The primary comparison uses $k = 6$ in both models; additional GPT-5.5 runs use $k = 3$ and $k = 9$. Retrieval does not modify model parameters.

**The implemented lexical score.** For term $t$ with document frequency $\mathrm{df}(t)$ in $N = 12$ events, the implementation uses

$$\mathrm{IDF}(t) = \log\left(1 + \frac{N - \mathrm{df}(t) + 0.5}{\mathrm{df}(t) + 0.5}\right). \tag{4}$$

This term is positive even at $\mathrm{df}(t) = N$. Word tokenization therefore does not literally set every ubiquitous query term's weight to zero. Instead, it fails to bind a value to the requested dimension, while word frequency can vary with many irrelevant dimensions. Field-pair tokenization preserves that binding. Ties are resolved by original record order. The equality structure and fixed record format make this result especially amenable to an exact filter or a relational index; the comparison does not establish a new ranking method.

Table 11: All GPT-5.5 depth settings on shifted queries. Every row contains the same 40 histories. Accuracy is a percentage; precision and recall are fractions.

| $k$ | Retriever | Accuracy | Precision | Recall |
|---|---|---|---|---|
| 3 | Field-pair BM25 | 85.0 | 1.000 | 0.612 |
| 3 | Word BM25 | 60.0 | 0.383 | 0.231 |
| 3 | Dense embedding | 50.0 | 0.458 | 0.271 |
| 6 | Field-pair BM25 | 97.5 | 0.825 | 0.954 |
| 6 | Word BM25 | 75.0 | 0.442 | 0.503 |
| 6 | Dense embedding | 70.0 | 0.483 | 0.558 |
| 9 | Field-pair BM25 | 100.0 | 0.589 | 1.000 |
| 9 | Word BM25 | 77.5 | 0.467 | 0.798 |
| 9 | Dense embedding | 85.0 | 0.469 | 0.807 |

**Conditional accuracy.** For each model at $k = 6$, all 64 decisions with complete recall are correct. Among 176 incomplete-recall decisions, GPT-5.5 answers 127 correctly and GPT-5.4 answers 128 correctly. Across all four model/depth files, there are 249 complete-recall decisions, all correct, and 711 incomplete-recall decisions, of which 499 are correct. Restricting to shifted queries gives 136/136 and 241/344, respectively. These are conditional descriptive counts, not independent Bernoulli trials. The complete-recall observation cannot identify the cause of every remaining error, nor does it imply that incomplete recall necessarily changes the sign of an offer's value.

**What is not recorded.** The archive does not retain exact selected dense-retrieval record identifiers or the decision's numeric estimate in the retrieval files. Dense selection cannot be reconstructed exactly. The deterministic lexical selections can be recovered, allowing the action-level audit below; missing numeric estimates still prevent a test of the model's exact arithmetic.

### D.1 Deterministic selected-evidence audit

We rerun only the saved lexical ranking functions over the regenerated, original histories. The generator seed and tie-breaking order are unchanged. For all 640 expected/shifted lexical records, reconstructed precision, recall, and the full-history oracle match the archive. Table 12 restricts the decision comparison to the 320 shifted-query records. It classifies the selected relevant rewards' sum as nonzero with the same sign as the full history, nonzero with the opposite sign, or zero. "Subset agreement" compares the archived action with the sign of this selected sum, excluding zero sums. Zero is neither positive nor negative, so it has no unique specified action under the original contract.

Table 12: Lexical evidence audit; 40 shifted-query records per row. "Same" and "opposite" compare selected and full-history signs. All 302 nonzero selected sums agree with the model's action, including 44 opposite-sign errors. Nine of the 18 zero-sum cases are correct against the full history.

| Model | $k$ | Index | Same | Opposite | Tie | Subset agreement |
|---|---|---|---|---|---|---|
| 5.5 | 3 | Field-pair | 34 | 6 | 0 | 40/40 |
| 5.5 | 3 | Word | 20 | 12 | 8 | 32/32 |
| 5.5 | 6 | Field-pair | 38 | 0 | 2 | 38/38 |
| 5.5 | 6 | Word | 30 | 10 | 0 | 40/40 |
| 5.5 | 9 | Field-pair | 40 | 0 | 0 | 40/40 |
| 5.5 | 9 | Word | 28 | 6 | 6 | 34/34 |
| 5.4 | 6 | Field-pair | 38 | 0 | 2 | 38/38 |
| 5.4 | 6 | Word | 30 | 10 | 0 | 40/40 |

This audit was added on September 16, 2026, after inspecting the responses. The audit file records selected indices, partial and full sums, and source-code hashes. No decision response was regenerated. All rows reuse the same 20 base pairs, and complete-recall observations are a subset of these traces rather than independent supporting evidence. Agreement with a partial-sum oracle is an observable behavioral fact, not proof of the model's internal computation or a causal explanation for dense-retrieval errors.

For completeness, the original descriptive correlations across twelve retriever/model/depth cells are $r = 0.936$ for accuracy with recall and $r = 0.638$ for accuracy with precision. These cells share histories. At $k = 6$, the two models agree on 239 of 240 actions, but dense embeddings were recomputed per run, so identical selected records are not guaranteed.

## E HISTORY INTERVENTION AND ACTION ORDER

This separate archived control changes experience histories rather than the memory-writing cue. It is retained to document the original research question and its negative complete-contract result; it is not evidence for the query-shift claim. Two histories share reward order, total resources, final physical state, and available actions, but reverse which device has better observed outcomes. We estimate

$$D_m = P(A \mid s, m(H_A)) - P(A \mid s, m(H_B)). \tag{5}$$

Memory is absent, a model-written summary, or raw events. Under the incomplete contract, device values are unknown and the histories provide evidence. Under the complete contract, both devices independently return $+6$ or $-2$ with equal probability, giving the same known expected value of $+2$. Calls are stateless; paired no-memory prompts are byte-identical. This design tests context-mediated choice, not parameter learning or enduring preferences.

In each base pair, the eight rewards are four $+6$ outcomes and four $-2$ outcomes in a shared shuffled order. The favored device receives three positive and one negative outcome (sample mean $4$), while the other receives one positive and three negative outcomes (sample mean $0$). Swapping device assignments changes which action the history favors, while preserving the reward sequence. Device identifiers are randomized across pairs. The final state explicitly resets resource to 50, debt to zero, and pending effects to none. No prior reward changes that supplied state.

Memory summaries are prompted to use at most 80 words and constrained by a 600-character schema. Separate summaries are written for the complete and incomplete contracts. The decision objective is to maximize expected resource after one choice, and either action is permitted when the known values tie. The no-memory condition uses an identical absence message within each pair. Each model's original batch uses seed 20260821, 32 base pairs, and base offset 100. The counterbalanced batch uses seed 20260822, 32 new pairs, and offset 200. Each batch has four decision samples per history/contract/memory cell. In the counterbalanced batch, prompt and action-schema order alternate together across base pairs and remain fixed within a pair.

Table 13: Counterbalanced history effects $D_m$. Each effect is averaged over 32 independent base pairs, with four samples per cell.

| Model | Contract | No memory | Summary | Raw events |
|---|---|---|---|---|
| 5.5 | Incomplete | 0.000 | 1.000 | 1.000 |
| 5.5 | Complete | 0.000 | 0.016 | 0.023 |
| 5.4 | Incomplete | 0.000 | 1.000 | 1.000 |
| 5.4 | Complete | 0.000 | 0.008 | 0.000 |

In the original, fixed-order batch, complete-contract raw-event effects are 0.0156 for GPT-5.5 and 0.0000 for GPT-5.4. Both models again produce effects of 1.0000 under the incomplete contract and zero without memory. Counterbalancing makes the aggregate complete/no-memory choice rate exactly $P(A) = 0.5$ in both models, while keeping the paired history contrast identified. It does not prove that the model samples uniformly when two actions appear in a single prompt: local first-option preferences can persist.

All complete-contract decisions in the four files report the correct expected reward of $+2$ for both devices. In the counterbalanced batch, the GPT-5.5 complete/raw effect is $0.0234375$, with exact sign-flip $p = 0.25$ and a 100,000-resample base-pair interval $[0, 0.0546875]$ (seed 20260820). The incomplete/raw effect is 1.0000 with $p = 4.66 \times 10^{-10}$ for each model. These results do not meet the registered large complete-contract effect in this setting, while preserving the positive control for useful historical evidence. Failure to meet that gate does not establish a universal absence of experience-dependent behavior.