SEA-BOUNDARY: A DIAGNOSTIC BENCHMARK FOR STATE PREPARATION, INFORMATION ACCESS, AND EVIDENCE USE IN LONG-CONTEXT LLM SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Limited context capacity motivates studying how LLM systems save information for later use and return to original documents when more detail is needed. We study how these systems prepare reusable memory (**State**), use it to guide requests for information from the documents (**Access**), and answer using State together with the retrieved passages (**Evidence**). When such a system answers incorrectly, the answer alone does not reveal whether the problem arose in State preparation, information access, or evidence use. We introduce SEA-Boundary, a controlled diagnostic benchmark for how knowledge prepared before questions guides the selection and interpretation of additional document facts. We specify what knowledge a model should retain and generate 96 synthetic document collections across four task families, deriving their questions and updates from the same facts and rules. Automated checks verify which combinations of retained knowledge and source records determine each answer, allowing new examples to be generated and checked. We compare independent function tests with complete workflows and inspect the memory and passages actually supplied in selected cases. Experiments show that correctly answering questions about State does not guarantee retrieving the facts needed to use that knowledge. Content checks identify cases where the relevant knowledge is correct in State but the model requests incomplete or irrelevant information. Inspecting actual answering inputs further distinguishes missing or incorrect information from mistakes made despite sufficient information. These tests evaluate memory through what it enables the model to read and answer, and help locate whether preparation, information access, or evidence use needs improvement.

1 INTRODUCTION

We study how large language models (LLMs) build reusable memory from long documents and use it to find information and answer later questions. Finite context capacity and the cost of processing full documents motivate keeping a shorter memory. When this memory is prepared before questions are known, it cannot be assumed to contain every detail needed later. The model can instead use each question and its memory to decide which original passages to read, an approach explored by ReadAgent and MemoRAG (Lee et al., 2024; Qian et al., 2025). It can reuse the memory across questions about the same documents (Li et al., 2025) and update it when source information changes.

We call the retained memory **State**, the model's request for source information **Access**, and the original passages returned by executing the request **Evidence**. SEA abbreviates State, Evidence, and Access. This gives three functions to evaluate: State preparation, information access, and evidence use (Figure 1). The model answers using both State and Evidence. Coding agents offer a practical analogue: they retain compact context (State), request relevant file content (Access), and use the returned code passages (Evidence) to carry out coding tasks (MacCallum & Fioca, 2026). State helps the model decide what to read and how to use the returned information. An error in memory can therefore lead to reading the wrong passages, while an answering error can occur even when all required information is available. Final-answer accuracy alone does not distinguish these failures or identify which function needs improvement.

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.

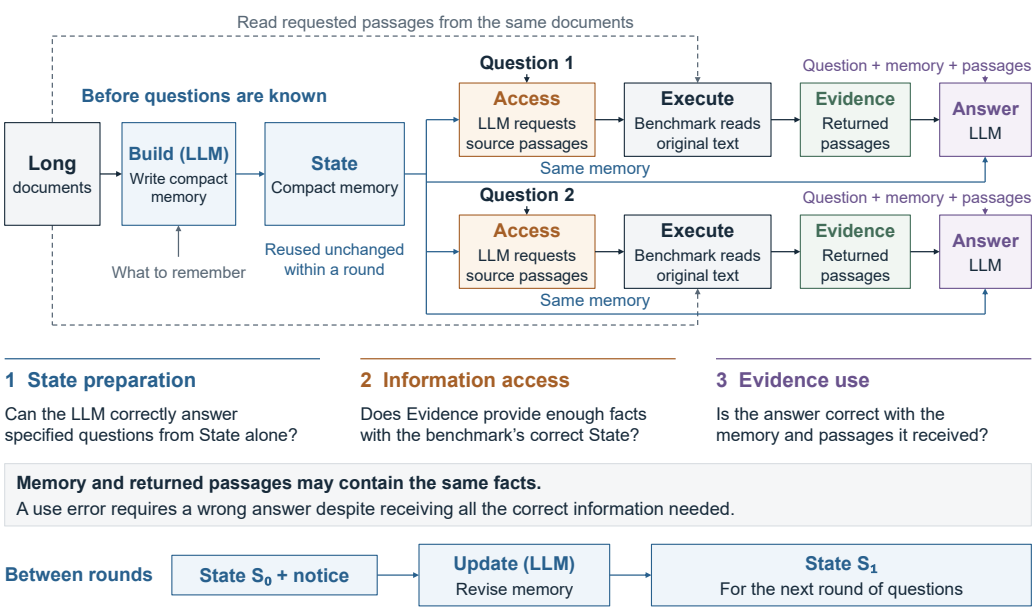


Figure 1: **Preparing memory, reading source passages, and answering.** State guides source reading and answering, is reused within a round, and is updated between rounds. State-only tests branch from State without source reading; independent tests supply correct State or Evidence in place of model-produced inputs. State and Evidence may overlap. Section 3 defines these tests and the checks needed before attributing an answering error to evidence use.

Existing memory diagnostics compare stored, retrieved, and reference information (Cheng et al., 2026; Yu et al., 2026). Retrieval and generation diagnostics check whether retrieved passages contain the information needed to answer, and whether the model uses it successfully (Ru et al., 2024; Joren et al., 2025). We ask whether memory prepared before questions are known helps the model find and use the right source information. A fact absent from memory need not indicate a preparation failure if it should be read later; retrieving that fact is not enough if memory lacks the rule needed to use it. To evaluate this connection, we must specify what knowledge the model should retain and which additional facts it needs for each answer.

We introduce **SEA-Boundary**, a controlled diagnostic benchmark named for the observable information interfaces between these functions. Each task connects knowledge prepared before questions, later source selection, and answering. State-only questions test whether the model can report or apply the required knowledge without rereading the documents. Source-dependent questions test whether it can combine State with additional document facts. We test the complete workflow using the model's own memory and requests, and test individual functions with the correct memory or passages supplied directly. For inspected cases, we check whether the memory and passages actually given to the model contain the correct information needed to answer, including any extra facts saved in State. Selected correction tests check whether supplying missing facts or correcting wrong information also fixes the answer.

We generate source documents, questions, updates, and reference answers from sampled facts and rules, allowing new examples to be generated reproducibly. Update notices let us test whether the model revises its memory while preserving knowledge that should remain unchanged. We check each example by computing the reference answers from the sources and testing the effects of removing or changing the information needed to answer. The evaluation covers four task families with 96 generated document collections, each containing approximately 5,800–21,900 tokens of source text. To check that the task format also captures relationships found in natural text, we construct an additional set of tasks from unchanged HotpotQA passages.

Experiments show that models can correctly answer a State-only question yet fail to retrieve the facts needed for a source-dependent question that uses the same knowledge. In some inspected cases, the relevant knowledge is correct in State, but the model's request leaves out required records or selects the wrong ones. Memory evaluation must therefore examine how the model uses retained knowledge to find information, beyond whether it can answer questions about that knowledge. Independent tests also obtain correct answers to questions for which the full workflow never reached the answering stage. Conversely, in separate correction tests on selected tasks, supplying missing facts or correcting wrong information does not eliminate every answering error. Together, these results show why checking what the model remembers, requests, and receives is necessary to interpret its final answer score.

Contributions. **(1) Tasks with checkable information dependencies.** Reproducible generators specify how knowledge retained before questions determines which source facts to read and how to use them, with tests across updates. **(2) Evaluation across these dependencies.** Independent tests and workflows compare readable knowledge with successful selection; actual-input inspections distinguish inadequate information from errors in its use. **(3) Empirical findings.** We identify selection failures despite correct task-relevant State and show why answering failures with different information defects require different checks and repairs.

2 Related Work

2.1 Memory and Guided Reading

Prepared memory can guide later reading: ReadAgent uses gist memories to select original passages (Lee et al., 2024), while MemoRAG generates retrieval clues from global memory (Qian et al., 2025). MemGPT manages movement between memory tiers (Packer et al., 2023). These systems use different forms of memory to support later reading; our implementation exposes textual State for inspection. The survey by Hu et al. (2026) discusses State, Evidence, and Access as functional distinctions. These systems motivate evaluating prepared knowledge through its contribution to later source reading.

2.2 Long-Context Evaluation

Existing benchmarks cover understanding, reuse, and changing knowledge. Understanding long inputs is tested by LongBench and HELMET (Bai et al., 2024; Yen et al., 2025); shared-context processing and cache reuse are examined by SCBench (Li et al., 2025). For changing knowledge, LongMemEval tests temporal reasoning and updates (Wu et al., 2025), while StreamMemBench follows information through feedback and later tasks, checking memory additions and revisions (Liu et al., 2026b). SEA-Boundary follows prepared knowledge into later source selection and answering, both before and after explicit updates. The evaluation asks how preparation, access, and use contribute to completing the same task.

2.3 Diagnostic Evaluation

Existing diagnostics already separate memory functions. WhenLoss compares source evidence, stored memory, and retrieved memory with a fixed reader (Yu et al., 2026). AMemGym generates conversational tasks from structured user states and probes writing, reading, and use, including answering from correct supplied information (Cheng et al., 2026). InMind distinguishes direct recall from applying implicitly relevant memories (Li et al., 2026). EvalMem jointly examines stored facts, native retrieval, and answering from oracle evidence, and includes a generated dynamic benchmark (Liu et al., 2026d).

For retrieval and answering, RAGChecker measures fact coverage and use separately (Ru et al., 2024), while Sufficient Context distinguishes inadequate information from errors despite sufficient information (Joren et al., 2025). RECON generates questions with explicit proof dependencies and checks coverage of their required evidence (Arya, 2026); Hindsight Memory-PRM tests answers after memory-entry deletion to guide memory management (Jia et al., 2026).

SEA-Boundary focuses on how knowledge retained before a question determines which additional source facts to request and how to interpret them. For example, a model may correctly retain which suppliers to assess and what criteria apply, yet request records for only three of four suppliers (Section 5.3). The task definition lets us check the retained relationships, requested records, and joint support from State and Evidence separately. State may omit a needed rule or supply a fact missing from retrieved passages. Our generators preserve these dependencies when producing new examples, so preparation, selection, and answering can be evaluated within the same task. Appendix F details the comparisons.

3 BENCHMARK DESIGN

SEA-Boundary tests whether knowledge retained before questions are known helps a model find and use information later. We inspect the information passed between functions: State is available to both Access and Answer, and Access determines the Evidence given to Answer. These observable interfaces are the *boundaries* named in the benchmark.

3.1 PREPARATION, ACCESS, AND ANSWERING

Model configuration M first receives documents $\mathcal{D}$ and public instructions P specifying the knowledge to retain. Build produces textual State S before any questions are revealed. For each later source-dependent question q_i^t, Access requests information and Answer uses the returned Evidence E_i together with State:

$$\begin{aligned} S &= \mathrm{Build}_M(\mathcal{D}, P), & A_i &= \mathrm{Access}_M(S, q_i^t), \\ E_i &= \mathrm{ResolveRead}(\mathcal{D}, A_i), & \hat{y}_i &= \mathrm{Answer}_M(S, q_i^t, E_i). \end{aligned} \tag{1}$$

We use typed requests to make the model's information choices explicit and inspectable. Access specifies subjects, relations, and scope, without answer values or document locations. ResolveRead (Execute in Figure 1) performs lookup and filtering to return original text. One request may contain multiple operations but cannot be revised after Evidence arrives: this tests selection from State and the question before retrieval feedback. Each function receives the public task instructions. Access and Answer use fresh calls without any earlier model answers, and State is passed verbatim, unchanged within a round.

Preparation instructions require rules and relationships and direct models to leave specified records in the documents. Extra retained facts incur no separate penalty and count as information available to Answer. We measure State/source length ratios separately from generation limits. A *reference State* contains the correct required knowledge; its length does not impose a size limit on model-written State.

3.2 QUESTIONS AND CONTROLLED UPDATES

Each required item of knowledge, or *commitment*, has a State-only question q_j^v. The model answers without reading the documents: $\hat{v}_j = \mathrm{Recall}_M(S, q_j^v)$. We call this State-only answering *readout*. Superscripts t and v distinguish source-dependent and State-only questions; i and j index questions of each type. Questions test relationships or rules and their application; reference State alone must suffice. Changing the tested knowledge must change the reference answer, while unrelated changes leave it stable. A correct response shows that the tested knowledge is usable through this reader. Assessing the State's contents requires the checks in Section 3.3 and Appendix C.1.

For each source-dependent question, we specify sets of document records that determine the required output together with reference State, including requested source facts and operational results. A set must be sufficient; it is also minimal if removing any record makes it insufficient. Multiple sets may qualify. These requirements let us check what Access retrieves, separately from whether the model answers correctly.

Between rounds, a supplied notice U_1 adds or revises knowledge: $S_1 = \mathrm{Update}_M(S_0, U_1, P)$. Update receives the actual previous State and complete notice, but no source archive or question–answer history, and returns a complete new State. Separate State-only questions test changed and unchanged knowledge; source-dependent questions test its later use. Evidence excludes the notice,

so its contents must remain available through State. We supply the update time and notice to test revision and preservation under controlled conditions.

3.3 Separating function performance from workflow outcomes

Workflows use the model's own State and requests. Independent tests supply correct inputs: reference State for Recall and Access, reference previous State for Update, and correct State plus sufficient Evidence for Answer. They test functions even when an upstream failure prevents the workflow from reaching them.

Access validity and execution are distinct from acquiring sufficient information. We distinguish four measurements:

Measurement	**What it establishes**
State-only readout	Whether a reader can answer about the supplied State; it does not certify State contents.
Evidence sufficiency with reference State	Whether retrieved records, with correct reference knowledge, determine the required report.
Actual joint-input sufficiency	Whether the supplied State and Evidence together contain correct, applicable information for that report.
Answering with verified inputs	Whether Answer succeeds after actual joint-input sufficiency has been established.

Only a normally completed, format-valid answer with a semantic error under sufficient actual inputs and unambiguous requirements is a use error. These checks cover selected cases. Original-run inspections combine deterministic record checks with one auditing LLM agent per case, which reads State and Evidence with outcomes visible (Appendix C.1). Selected correction tests replace incorrect or missing information in State, Evidence, or both at Answer, holding other inputs and generation settings fixed; they do not rerun Access. Appendix A.4 details the scoring distinctions.

4 Benchmark Construction and Validation

To implement these tests, we need to know which knowledge a model should retain, which document facts each question needs, and how they determine the answer. We generate these together and check their consistency.

4.1 Generating documents and paired tests

We first sample a structured description of entities, relationships, rules, source records, and changes, called a *world*. From it we produce documents, preparation instructions, notices, and reference States (Figure 2). We fix the world and documents before deriving questions and answers. Data generation is independent of model outputs and generation budgets.

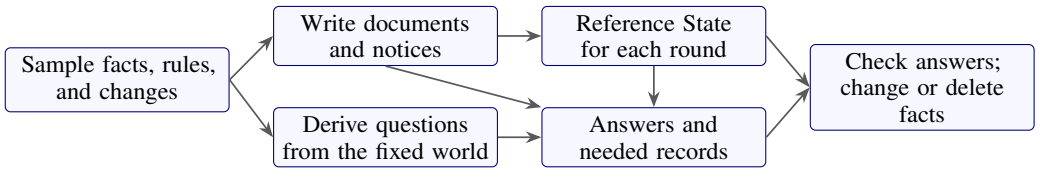


Figure 2: **Generating tasks with known information requirements.** Documents and reference State come from the same sampled facts and rules. Questions are derived afterward. Checks test answer consistency, information needed to determine answers, and acceptance of equivalent records.

Example (simplified). State assigns a coordinator to the central line and specifies three valve sets per kit. A question asks about Batch A. Access therefore requests central-line stock for Batch A: the line comes from State, the batch from the question. Answer uses the retained recipe to convert stock into kits. Appendix A.1 gives the full example and update.

The four families pair purchasing policies with quotations and stock, assembly recipes with component records, qualification rules with scores, and service priorities with offers. For each required State item, we generate a State-only question. Each initial or updated State also serves three source-dependent questions, for which we identify the records needed alongside reference State. Notices specify changed and stable knowledge for paired tests across rounds. Appendix A gives family statistics and a worked example.

4.2 Validation and additional examples

Automated checks test three properties. *Answer consistency:* public documents and notices support the reference States and answers; separate replay reconstructs all 192 States and checks 1,182 State-only and 576 source-dependent questions. *Information requirements:* record-deletion checks test whether the remaining records support the required report; interventions on relevant rules or relationships test whether the answer changes. Appendix A.3 specifies the family-specific checks and their relation to necessity proofs. *Equivalent information:* valid alternative supports and relocated records remain accepted; records from other worlds are rejected (Table 5). Selected agent-assisted reviews also check that complete documents and instructions express the intended task. Section 5.4 examines ambiguities in the units and formulas of intermediate answer fields.

Within each implemented family, new random seeds generate additional examples subject to the same checks. A new family requires its own rules, answer computation, and document-writing procedure under the shared protocol.

5 Evaluation and Results

Evaluation setup. We evaluate 96 document collections, 24 per family, with 1,182 State-only and 576 source-dependent questions per configuration across two rounds. We retain original responses and output limits. *Complete-answer reference match* requires normal completion, the requested format, and every value matching the reference. A separate *task-result match* excludes only specified intermediate arithmetic values; format, identities, source facts, and final results must still match (Appendix C.3). Both scores require all fields; the fixed field map is an additional analysis, not a rerun. Actual-input sufficiency is checked separately; uncertainty resamples initial worlds (Appendix B.3).

Model configurations. We use thinking for stronger reasoning models and non-thinking settings for smaller models to limit prolonged, repetitive generation. K3 denotes Kimi K3; DeepSeek low/max denote DeepSeek V4 Pro at low/maximum reasoning effort. Appendix B gives settings and additional small-model tests. Repeated-call and sensitivity controls appear in Appendices C.4 and E.

5.1 How usable is prepared and updated State?

K3 correctly answers 571/576 initial State-only questions and all questions about revised or added knowledge (Table 1). Qwen3.8-27B's stable-knowledge readout falls from 494/510 to 477/510, while reference-State readout rises from 503/510 to 507/510. Readout combines State preparation and reader behavior: DeepSeek max's score falls even with reference State (Appendix B.2). Median State/source ratios are 10.70%–19.11% among completed outputs containing State text, without a shared State-size limit.

5.2 What do independent tests reveal beyond workflow outcomes?

Independent tests reveal answering ability hidden by missing workflow calls: with correct inputs, K3 matches 35/35 and DeepSeek max 151/159 tasks without an original Answer (Appendix B.1). Table 2 separates recorded responses from saved-request replay. Across seven workflows, 50/4,032 planned tasks have replay-executable requests but no original Answer; replay does not fill these missing calls (Appendix B.1).

Executable requests need not acquire sufficient information. With reference State, Qwen3-30B-A3B has sufficient Evidence on 51/281 requests executable in replay, versus 66/116 for Falcon-H1-7B. Qwen's purchasing and assembly requests fail the public interface; qualification and service requests can execute yet omit required facts.

Table 1: **State size and knowledge usable before and after updates.** Size is the median State length as a percentage of source-document tokens, over n completed outputs containing State text out of 192 planned. We use the same token-counting method for every model (Appendix A.2). Other columns count correct State-only answers: Revised/Added test changed/new knowledge; Stable is before $\rightarrow$ after Update.

Configuration	State/source % (n)	Initial (of 576)	Revised (of 66)	Added (of 30)	Stable (of 510 each)
Thinking enabled					
K3	10.70 (190)	571	66	30	$506 \rightarrow 507$
DeepSeek low	12.28 (192)	563	62	29	$498 \rightarrow 501$
DeepSeek max	15.40 (181)	557	57	28	$494 \rightarrow 466$
Qwen3.8-27B	19.11 (175)	558	59	30	$494 \rightarrow 477$
Without thinking					
Qwen3-30B-A3B	14.75 (168)	251	21	7	$230 \rightarrow 234$

Table 2: **Recorded responses and a separate request-replay audit.** Original limits; 1,182 readout and 576 Access/Answer tasks per condition. A scores recorded responses (complete-answer and additional task-result matches). B replays saved requests; sufficiency uses reference State. Ref./Model denotes reference/model-produced inputs. Independent Answer receives correct inputs regardless of Access.

A. Recorded model responses

		State-only	Answer		
Configuration	Inputs	Matches	Calls	Complete	Task result
K3	Ref.	1179	576	576	576
	Model	1174	541	539	539
DeepSeek low	Ref.	1166	576	562	573
	Model	1155	483	471	482
DeepSeek max	Ref.	1111	576	525	536
	Model	1108	417	404	410
Qwen3.8-27B	Ref.	1168	576	569	575
	Model	1124	536	518	520
Qwen3-30B-A3B	Ref.	674	576	102	118
	Model	513	276	28	36

B. Saved-request replay

	Reference-State requests		Model-State requests	
Configuration	Executable	Sufficient	Executable	Sufficient
K3	567	565	550	547
DeepSeek low	512	505	512	511
DeepSeek max	391	391	422	420
Qwen3.8-27B	562	562	543	541
Qwen3-30B-A3B	281	51	276	73

Resource limits also matter. Across recorded actual and independent Access calls, DeepSeek max has 296/1,140 length stops, versus 4/1,152 for low effort under the same ceilings (Appendix B.3). These are configuration outcomes, not isolated comparisons of semantic ability.

5.3 Does knowledge in State guide successful source selection?

Four inspected Qwen3-30B-A3B tasks fail source selection despite correct relevant State. In one qualification example, State records "Reserve buyer → South zone" and "South zone → South review panel." The task requires twelve scores: four suppliers on three criteria, including a waived criterion. Access requests Alder, Birch, and Cedar; Evidence returns their nine scores but omits Dover's three. The existing review found correct relevant State without compensating scores. A second question repeats the omission with identical State; two other cases select the wrong department (Appendix C.4).

Pairing update-dependent tasks with State-only questions locates these cases (Figure 3). Qwen answers 56/192 correctly; of these, 34 requests execute in replay and 11 are sufficient. We inspect all 23 executable but insufficient cases. Beyond the four above, 19 have State defects whose causal role is unestablished. The broader cohort supports only a readout–selection discrepancy (Appendix C.4). For the four stronger configurations, every replay-executable request in its selected subset is sufficient.

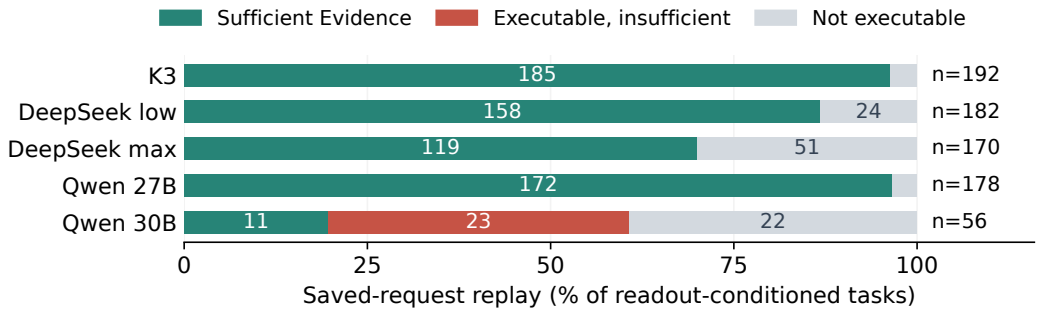


Figure 3: **Source selection after a correct State-only answer.** Original limits; n is the subset with correct readout among 192 update-dependent tasks. Access receives the same State, without the readout answer. Bars describe saved-request replay, not original execution events; sufficiency uses reference State. Qwen 27B/30B denote Qwen3.8-27B/Qwen3-30B-A3B.

Natural-text application. We express unchanged HotpotQA passages in the task format: six source groups, 23 film–director questions, one template, no Update (Yang et al., 2018). DeepSeek's workflow gives 23/23 source-supported answers (21/23 format-compliant). A separate reference-State control gives 4/23 supported answers with question-only BM25 and State at Answer, versus 21/23 using State to form the query, both limited to two paragraphs. Sources contain 557–909 tokens; Appendix D gives closed-book and additional retrieval controls.

5.4 What explains answers marked incorrect?

The same answer mismatch can require different repairs. One K3 State identifies the wrong production line: Access retrieves that line's genuine prices, and Answer correctly sums them to 15 per kit rather than the required line's 24. The binding persists across an unrelated update. In contrast, one answer uses another service's lead time despite receiving the correct offer. Checking actual State and Evidence distinguishes these input and use defects (Appendix C.4).

We apply this inspection to all 39 normally completed, format-valid complete-answer mismatches in K3, DeepSeek low/max, and Qwen3.8-27B (Figure 4). Fifteen lack correct and sufficient actual inputs; ten of these appropriately abstain. Five misreport facts or abstain despite sufficient inputs. Nineteen answers have correct task results but differ on intermediate values: 18 on assembly capacities and one on a qualification margin. The original instructions did not separately define the capacity unit or margin formula. These remain mismatches under the reference scoring convention, not established semantic use errors. Table 2 reports both scores; the same field map applies to all configurations. A separate clarification documents these conventions (Appendix C.3).

Agent-assisted inspections cover 113 selected cases among 2,358 nonempty original Answers (Appendix C.2); this includes the complete 39-case cohort.

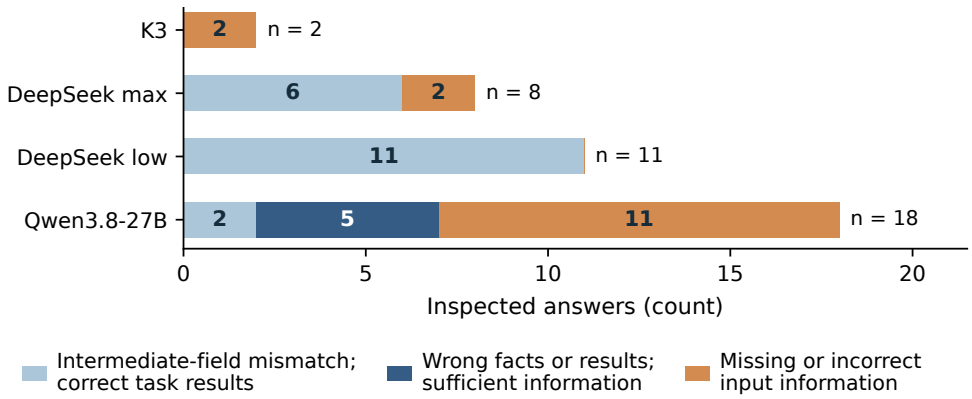


Figure 4: **Explaining the 39 complete-answer mismatches.** All completed, format-valid mismatches in four configurations. Intermediate-field mismatches have correct task results but ambiguous field definitions.

On eight selected DeepSeek tasks without thinking, correcting inputs without rerunning Access improves matches from 0/4 per group to 4/4 (Evidence-only) and 3/4 (joint). Joint correction reaches 4/4 with maximum thinking and a larger allowance. Baselines use the same structured format (Appendix C.5, Table 14).

6 Discussion and Limitations

We test textual State, one request before Evidence, and supplied update notices. Extra retained facts count toward support; no shared State-size limit constrains compression. Input audits cover selected cases; readout measures usability through a reader. K3 approaches ceiling in several families. Natural-text evidence covers short HotpotQA sources, one template, and no Update. Skipping retrieval, autonomous update timing, controlled length scaling, and deployed systems remain outside the evaluation.

Full-source answering largely catches up under higher resources on 48 matched tasks (Appendix E.3). Costs cover answering only, excluding preparation, indexing, and retrieval; resource follow-ups and repeated calls remain separate from original runs.

Future directions. An immediate extension is to evaluate when incoming information or a query should trigger State revision or a refresh from Evidence, including unnecessary writes and delayed updates. Generated retention requirements, support sets, and reference answers provide targets for studying supervision of individual functions. We plan to use these targets to develop learned internal State and access mechanisms without serializing intermediate memory as text, adapting diagnostic tests to their observable behavior. The longer-term objective is to improve the quality–efficiency frontier relative to full-attention Transformers as context length and task load grow. Evaluation would measure quality alongside preparation, updates, source access, and answering costs, accounting for reuse across questions.

7 Conclusion

SEA-Boundary evaluates State preparation, information access, and evidence use within long-context LLM systems. Its generated tasks connect the knowledge models are asked to preserve to the source facts needed for later answers, with checks that validate these relationships across updates. Experiments show why evaluating retained knowledge requires more than testing recall: a model can answer questions about State correctly yet fail to select the information needed for a subsequent task. Independent function tests and checks of actual answering inputs help locate what an answer score leaves unresolved. This makes preparation and information access assessable together, while keeping error attribution within the scope of verified evidence.

AI USE STATEMENT

The authors developed the core research concept and directed the research. LLM assistants, including Codex and ChatGPT, supported refinement of the argument, experimental design, synthetic-data generation and generator development, implementation and review of evaluation code, analysis and interpretation of results, literature search and organization, and manuscript drafting, revision, and internal review. An image-generation tool assisted early figure design; the current overview is redrawn with editable PowerPoint objects, and other figures also have editable sources. Data validation and evaluation checks are described in Section 4 and Appendices A and C. The authors take responsibility for the final text, claims, data, code, and figures.

ETHICS STATEMENT

The main benchmark uses deterministic synthetic worlds. A separate adaptation uses public Wikipedia paragraphs distributed with HotpotQA; no human-participant study was conducted. Appendices A and D describe construction and source selection. The adaptation uses the HotpotQA-distributed text under its CC BY-SA 4.0 data license. Systems that retain or reopen source material can create privacy and safety risks in deployment, requiring a separate assessment of data governance and access control.

REPRODUCIBILITY STATEMENT

Sections 3–4 specify the interface, updates, and construction checks. Appendix A details data, input visibility, scoring, and statistical units; Appendix B gives model settings and generation allowances. Appendix C describes actual-input inspections and controlled corrections. Appendices D and E specify natural-text and sensitivity studies, including their samples, resource conditions, and response reuse. The anonymous data package contains the frozen benchmark, generation and prompt code, and offline reference-chain checks. A separate evidence supplement contains workflow correspondences and six complete inspected cases. An addendum provides paired correction responses and a replayable necessity witness. Versioned additions index the six cases, verify saved-request replay, and clarify answer-field units and formulas while preserving evaluated prompts. These materials support checking data and selected interpretations; they do not reproduce all model scores or the auditing agent's semantic judgments.

REFERENCES

Mihir Shriniwas Arya. RECON: Benchmarking agent memory for compositional reasoning over long contexts, 2026. URL https://arxiv.org/abs/2607.16716v1.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Aviv Bick, Tobias Katsch, Nimit Sohoni, Arjun Desai, and Albert Gu. Llamba: Scaling distilled recurrent models for efficient language processing, 2025. URL https://arxiv.org/abs/2502.14458.

Jiayang Cheng, Dongyu Ru, Lin Qiu, Yiyang Li, Xuezhi Cao, Yangqiu Song, and Xunliang Cai. AMemGym: Interactive memory benchmarking for assistants in long-horizon conversations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=sfrVLzsmlf.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Xingbo Du, Loka Li, Duzhen Zhang, and Le Song. MemR^3: Memory retrieval via reflective reasoning for LLM agents, 2025. URL https://arxiv.org/abs/2512.20237v1.

Falcon-LLM Team. The Falcon 3 family of open models, 2024. URL https://huggingface.co/blog/falcon3.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, Anna Golubeva, Vasudev Shyam, James Whittington, Jonathan Pilault, and Beren Millidge. The Zamba2 suite: Technical report, 2024. URL https://arxiv.org/abs/2411.15242.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783v3.

Zhuangfei Hu, Runqing Xu, Yuchen Li, and Yongqi Zhang. Preserving memory function in hybrid language models: A survey of state, evidence, access, and granularity. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026. To appear.

Haoxuan Jia, Yang Liu, Yingguang Yang, Yancheng Chen, Chongyang Zhang, Hao Zheng, Qian Li, Yulin Huang, Jianshen Zhang, Yongzhi Qi, Shang Luo, Kefu Xu, Hao Peng, Junyu Lu, Du Cheng, Philip S. Yu, and Bin Chong. Hindsight Memory-PRM: Supervising memory management with auditable hindsight credit, 2026. URL https://arxiv.org/abs/2608.29605v1.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/33dffa2e3d2ab74a783d1a8c292f66d9-Abstract-Conference.html.

Kimi Team. Kimi K3: Open frontier intelligence, 2026. URL https://github.com/MoonshotAI/Kimi-K3/blob/main/k3_tech_report.pdf. Technical report.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26396–26415. PMLR, 2024. URL https://proceedings.mlr.press/v235/lee24c.html.

Ruizhe Li, Mingxuan Du, Benfeng Xu, and Zhendong Mao. Keep it InMind: Benchmarking the implicit-association blind spot in agent memory, 2026. URL https://arxiv.org/abs/2607.24368v1.

Yucheng Li, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. SCBench: A KV cache-centric analysis of long-context methods. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a540b17fb2295c736d5afd6c507acf66-Abstract-Conference.html.

Chengzhi Liu, Yuzhe Yang, Sophia Xiao Pu, Yepeng Liu, Lin Long, Yichen Guo, Nuo Chen, Zhaotian Weng, Elena Kochkina, Simerjot Kaur, Charese Smiley, Xiaomo Liu, James Zou, Sheng Liu, Yuheng Bu, Songyou Peng, and Xin Eric Wang. WorldMemArena: Evaluating multimodal agent memory through action-world interaction, 2026a. URL https://arxiv.org/abs/2605.29341v2.

Guanming Liu, Yuqi Ren, Hansu Gu, Peng Zhang, Weihang Wang, Jiahao Liu, Ning Gu, and Tun Lu. StreamMemBench: Streaming evaluation of agent memory for future-oriented assistance, 2026b. URL https://arxiv.org/abs/2606.14571. arXiv version 2; accepted to Findings of EMNLP 2026, as stated by the authors.

Xule Liu, Yijun Liu, Chao Li, and Shao Kun. D^{2}ACCI: A dual-loop diagnostic protocol for evidence-preserving agent memory, 2026c. URL https://arxiv.org/abs/2608.17756v1.

Zeyu Liu, Jian Zhong, Rongduo Han, Ziyang Wu, Shunye Tang, Chenghao He, Yaxuan Yang, Yihang Qiu, Ailing Wang, Xiao Liang, Guohuan Xie, Xiaokang Xue, Gongchen Li, Haining Zhang, and Wei Wang. EvalMem: An operation-level diagnostic framework for long-term memory systems, 2026d. URL https://arxiv.org/abs/2609.22231v1.

Xianxuan Long, Zhikai Chen, Shenglai Zeng, Shouren Wang, Kai Guo, and Jiliang Tang. MemTrace: Probing what final accuracy misses in long-term memory, 2026. URL https://arxiv.org/abs/2606.17328v1.

Noah MacCallum and Brian Fioca. Codex prompting guide. OpenAI Cookbook, February 2026. URL https://developers.openai.com/cookbook/examples/gpt-5/codex_prompting_guide. Accessed September 25, 2026.

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.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. MemoRAG: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM Web Conference 2025*. ACM, 2025. doi: 10.1145/3696410.3714805. URL https://doi.org/10.1145/3696410.3714805.

Peijun Qing, Fobo Shi, and Soroush Vosoughi. UTILMEM: Benchmarking evidence utilization in long-term conversational memory, 2026. URL https://arxiv.org/abs/2608.30508v1.

Qwen. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115v1.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Qwen Team. Qwen3.8-27B: Model card, 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B. Official model card. Accessed 26 September 2026.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Jiayang Cheng, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. RAGChecker: A fine-grained framework for diagnosing retrieval-augmented generation. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0692. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/27245589131d17368cccdfa990cbf16e-Abstract-Datasets_and_Benchmarks_Track.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://openreview.net/forum?id=pZiyCaVuti.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024. doi: 10.1145/3626772.3657878. URL https://arxiv.org/abs/2309.07597v5.

B. Y. Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. General agentic memory via deep research, 2025. URL https://arxiv.org/abs/2511.18423v1.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context models effectively and thoroughly. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f5332c8273d02729730a9c24dec2135e-Abstract-Conference.html.

Jiangnan Yu, Kisson Songqi Lin, and Jilong Wu. WhenLoss: Diagnosing write and retrieval bottlenecks in long-context memory systems, 2026. URL https://arxiv.org/abs/2605.24579.

Jingwei Zuo, Maksim Velikanov, Ilyas Chahed, Younes Belkada, Dhia Eddine Rhayem, Guillaume Kunsch, Hakim Hacid, Hamza Yous, Brahim Farhat, Ibrahim Khadraoui, Mugariya Farooq, Giulia Campesan, Ruxandra Cojocaru, Yasser Djilali, Shi Hu, Iheb Chaabane, Puneesh Khanna, Mohamed El Amine Seddik, Ngoc Dung Huynh, Phuc Le Khac, Leen AlQadi, Billel Mokeddem, Mohamed Chami, Abdalgader Abubaker, Mikhail Lubinets, Kacper Piskorski, and Slim Frikha. Falcon-H1: A family of hybrid-head language models redefining efficiency and performance, 2025. URL https://arxiv.org/abs/2507.22448.

Zyphra. Zamba2-7B-Instruct-v2: Model card, 2025. URL https://huggingface.co/Zyphra/Zamba2-7B-Instruct-v2. Official model card. Accessed 26 September 2026.

GUIDE TO THE APPENDIX

Appendix A supplies a worked task, data details, and evaluation rules; Appendix B gives additional function results; Appendix C examines selection and answering failures. Appendix D covers natural-text tasks, Appendix E tests instructions and resources, and Appendix F compares related work.

A DATA AND EVALUATION DETAILS

A.1 A WORKED ASSEMBLY EXAMPLE

Table 3 gives a reference solution for one synthetic assembly task. Names are shortened; questions and records are paraphrased. Before seeing questions, the system retains line assignments and assembly rules; batch-specific stock remains in the sources.

Table 3: **A worked task: retained rules and retrieved stock.** The example covers one coordinator, one kit type, and one batch. Quantities count clamp pairs, individual seal rings, and valve sets.

Part	Content
State	This coordinator's jobs go to the central assembly line. Each kit requires one clamp pair, one seal ring, and three valve sets. Deduct two units of each part type once, before assembly. Pack finished kits in pairs.
Question q^v	Which production line handles this coordinator's jobs? *Answer:* the central assembly line, using State alone.
Question q^t	Using the stock for Batch A, how many kits can we assemble for this coordinator? Show how many kits the stock of each part allows, then report the total, the number of full packages, and any kits left unpacked.
Access	Request the stock of clamp pairs, seal rings, and valve sets on the central assembly line for Batch A. The line and parts come from State; the batch comes from the question.
Evidence and calculation	Part / Source stock / After loss / Enough for (kits): Clamp pairs 9 7 7; Seal rings 14 12 12; Valve sets 13 11 3
Answer	The parts support 7, 12, and 3 kits, respectively. Valve sets limit production to **3 complete kits**: **1 full package** and **1 unpacked kit**.

The same stock under an updated rule. A notice reduces startup loss to one unit per part type; the recipe and packaging stay unchanged. The same stocks now support 8, 13, and 4 kits: **4 complete kits, 2 full packages, and none unpacked**. The notice is absent from the retrieved stock records, so the new loss rule must be retained in State.

A.2 SOURCES, QUESTIONS, AND UPDATES

Each generated world supplies a collection of documents, a retention policy, and an update notice. Source facts and their organization are fixed before evaluation questions are constructed. The sampler does not receive later questions, reference answers, required source records, or model budgets. Reference States, answers, and the records needed to answer are derived from this world. This order prevents generation from choosing source facts to fit a desired model answer.

Table 4 summarizes the evaluated corpus. Each of 96 worlds has two rounds and three source-dependent questions per round. Every required State commitment has a corresponding State-only question: 576 before Update and 606 afterward. Notices revise 66 commitments and add 30; 510

commitments should remain unchanged. These are changes to active policies, not a separate condition that merely reveals an existing fact. Reference States include the applicable authority, scope, and effective time, so an obsolete rule is distinguishable from its replacement.

Table 4: Dataset structure. Each family has 24 worlds and 144 source-dependent questions across two rounds. Source lengths count document bodies only; questions, answers, titles, and instruction wrappers are excluded. Support size counts records in an inclusion-minimal sufficient set.

Family	Source tokens	State-only questions	Revised / added commitments	Request operations	Support records
Purchasing	9,365–18,910	300	12 / 12	2–3	3–7
Assembly	10,086–10,157	294	18 / 6	1	3
Qualification	21,869–21,949	294	18 / 6	3	12
Service	5,797	294	18 / 6	1	1–2

We use the same tokenizer (`cl100k_base`) for document bodies, reference State, and model-written State. Initial reference States span 691–1,005 tokens and updated reference States 685–1,137 tokens. These lengths describe the reference representation; they do not impose a model State-size limit. Sources have no matched short/long versions, so their length variation is not a controlled test of context-length effects.

A.3 Validation and information requirements

Independent replay reconstructs reference State from the public sources and notice, executes the reference requests, and checks the resulting text and answers. Table 5 gives coverage for the evaluated corpus. Cross-world tests substitute each of the other 23 worlds for every recipient world and its six tasks: $24 \times 23 \times 6 = 3{,}312$ substitutions per family. They test whether unrelated records can wrongly satisfy the task, rather than merely whether a request executes.

Table 5: Reference-corpus checks. Counts are checked or rejected objects within the stated domain, with their own denominators.

Check	Purchasing	Assembly	Qualification	Service
Reference States reconstructed	48/48	48/48	48/48	48/48
State-only questions replayed	300/300	294/294	294/294	294/294
Reference task chains replayed	144/144	144/144	144/144	144/144
Accepted minimal supports	144	144	144	184
Single-member deletions checked	712	432	1728	264
Cross-world substitutions rejected	3312/3312	3312/3312	3312/3312	3312/3312

For a reference State and question, a source-record set is sufficient if their joint contents determine the required report under the public rules, including requested source facts and operational results. Qualification requires every provider–criterion score, including waived criteria; a qualification decision alone is not the complete requested output. It is inclusion-minimal if removing any member makes it insufficient. Multiple valid sets are accepted: 40 service tasks admit two alternatives, giving 184 supports for 144 tasks. Replacement and relocation checks distinguish the required facts from one particular record identity or address. Retained facts need not appear again in Evidence.

Each deletion check holds reference State and the question fixed, removes one record from an accepted support, and reruns a deterministic answer procedure. Purchasing follows the retained mappings to the required item lists, supplier or contract bindings, and numerical records. Assembly requires stock or prices for every recipe component. Qualification requires all twelve provider–criterion scores. These procedures check availability and scope of the facts needed to compute the complete report; they do not enumerate every possible missing value. Service instead enumerates capacity assignments allowed by the public rules and, when the selected service's waiting time is missing,

tests the endpoints of its legal range. It accepts a reduced set only when these completions give one unique report. Table 5 counts member deletions from accepted supports, including separate deletions for alternative supports.

For the first three families, a missing input to the answer procedure identifies a report dependency; proving necessity additionally requires that legal alternatives for the omitted information can change the report. For example, qualification certificates change one omitted score, validate the resulting world, and check that the report changes while the other facts remain fixed. Service certificates record disagreeing completions. Where a necessity claim is made, its witness consists of two legal worlds or update histories with identical remaining visible inputs but different answers. Visibility includes instructions, identifiers, source organization, and any information retained at the tested call. A failed model response after deleting a record is not itself proof that the record was necessary. The table therefore counts reduced-support checks; the associated necessity arguments and saved witnesses differ by family.

Example: why a missing service record matters. In this synthetic task, a coordinator must choose a transport service and report its capacity and the time from filing a request to departure. The selection rule is to take the first service on a priority list with at least six seats. State stores this rule, places a service named Haven first, and specifies six hours to process the booking. The source records give each service's capacity and the additional wait from booking confirmation to departure. The requested total time is therefore the six-hour processing time plus the selected service's waiting time.

To check whether Haven's record is needed, we remove it from Evidence, leaving State and the other services' records available. We then compare two legal versions of the missing record. In both, Haven has six seats and is the first qualifying service. In one, its wait after confirmation is one hour; in the other, it is twenty hours. All other facts remain unchanged. The required total is $6 + 1 = 7$ hours in the first version and $6 + 20 = 26$ hours in the second. The difference concerns a fact that the remaining inputs do not reveal, not a different selection rule.

Rebuilding State and the remaining Evidence in both versions produces identical inputs to Answer, including instructions, question, document identifiers, and text. Those inputs therefore cannot determine which total to report. This is the necessity check: two possible sources agree on everything Answer can see but require different outputs. The example comes from service world 44400, initial-round question 1; the supplementary addendum replays its stored check offline. This prose explains the original task rather than replacing its instructions, and the example does not establish proof coverage for every deletion test.

Complete-source agent reviews cover 26 distinct worlds. Twenty-four are selected by fixed ranks in assembly, qualification, and service batches; an eight-world control overlaps six and adds two purchasing worlds. The reviews use different checklists and complement executable checks; they are not 26 uniform human annotations or an agreement estimate.

A.4 Inputs and scoring

Build receives sources and the retention policy before seeing questions or notices. Each subsequent question uses a fresh context: earlier Recall answers and task answers are never added to State. Update receives the previous State and complete notice. Within a round, emitted State text is passed unchanged, including nonempty text stopped at the generation limit. Missing State or an unexecutable request can prevent a dependent call.

Independent tests supply reference State to Recall and Access, reference old State to Update, and reference State with sufficient Evidence to Answer. They proceed independently of workflow completion. Actual Answer receives the text selected by its own request, even if that text is insufficient. Access execution and Evidence sufficiency are scored separately; sufficiency in the function tables uses reference State. It does not certify the actual State or the joint inputs received by Answer.

Preparation instructions specify required mappings and rules with "Retain every" and discourage named current-record classes with "Do not retain." Evaluation imposes no separate penalty for retaining extra facts; they count toward actual answering support. Output allowances limit generation, including reasoning, rather than define a uniform memory capacity. Appendix E.2 examines these instructions under fixed allowances.

Complete-answer success requires normal completion, the public output format, and all requested fields to match. A separate task-result measure in Appendix C.3 isolates intermediate arithmetic fields without changing that success criterion. Source facts remain required under both scores; Appendix C.3 preserves the original intermediate-field ambiguities. Format violations, length stops, absent calls, and transport failures are not semantic use errors. Initial worlds are the statistical units: questions and rounds sharing a world are not independent samples. Planned tasks, returned answers, and input-verified subsets retain their own denominators.

B Additional Function Results

B.1 Model configurations and generation allowances

K3 (Kimi Team, 2026) uses the service's default maximum-thinking setting: the `k3` request specifies temperature 1.0 and omits explicit thinking/effort fields. DeepSeek V4 Pro (DeepSeek-AI, 2026) uses `deepseek-v4-pro`, thinking enabled at low or maximum effort, and provider-default temperature. Both efforts share the allowances in Table 6; reasoning and final output share each limit.

Qwen3.8-27B (Qwen Team, 2026) uses the Hugging Face checkpoint in BF16, a configured 262,144-token context, thinking enabled at xhigh effort, temperature 1.0, top-p 0.95, and top-k 20. It uses the DeepSeek allowances. Qwen3-30B-A3B-Instruct-2507 (Qwen Team, 2025) uses greedy decoding without thinking, a 32,768-token context, and the K3 allowances. The additional smaller models also use greedy decoding without thinking and the K3 allowances. Their context limits are 131,072 (Llama-3.1-8B-Instruct (Grattafiori et al., 2024)), 65,536 (Falcon-H1-7B-Instruct (Zuo et al., 2025) and Falcon3-Mamba-7B-Instruct (Falcon-LLM Team, 2024)), 32,768 (Qwen2.5-1.5B-Instruct (Qwen, 2024)), 16,384 (Zamba2-7B-Instruct-v2 (Glorioso et al., 2024; Zyphra, 2025)), and 4,096 (Llamba-8B (Bick et al., 2025)). These configurations do not isolate model size or architecture.

Table 6: Generation limits for complete workflows and independent function tests. DeepSeek's two effort settings and Qwen3.8-27B thinking share the DeepSeek rows.

Model	Setting	Build	Update	Recall	Access	Answer
K3	Purchasing	8192	4096	2048	4096	8192
	Assembly	4096	4096	2048	4096	2048
	Qualification	4096	4096	3072	4096	3072
	Service	4096	4096	4096	4096	4096
DeepSeek	Purchasing	18432	18432	16896	17408	16896
	Assembly	18944	18944	17152	17408	17152
	Qualification	19456	19456	17920	17920	17920
	Service	17920	17920	16896	17152	16896

Answer coverage. All recorded Answer calls return nonempty text for K3, DeepSeek low, Qwen3.8-27B, and Qwen3-30B-A3B. DeepSeek max returns nonempty text on 539/576 independent calls and 412/417 actual calls. Nonempty text does not imply normal completion or a correct answer. Access scores and recorded Answer counts are reported independently: an accepted request does not imply that an Answer call exists for its returned records.

We distinguish original execution from offline replay across 4,032 planned tasks. All 2,363 recorded Answer calls (2,358 nonempty) are replay-executable; none is rejected or unresolved in replay. Direct replay reproduces every original Answer's Evidence and ordered source records. For 313 purchasing cases, Python tuples serialize to the same JSON arrays; no record differs. Another 50 requests execute in replay but had no original Answer: nine K3, 29 DeepSeek low, five DeepSeek max, and seven Qwen3.8-27B. All 50 archived prompts permit the relevant literal arrays, and all 100 scalar-expansion schema checks pass. Original records show Access format rejection followed by an unsent Answer. Replay does not create original Answer observations.

The supplementary provenance audit provides exact prompts and outcomes, task-level Evidence comparisons, and the replay source manifest. The replay snapshot differs from some supplied

implementation files; its identity is reported separately. Historical validator/resolver source revisions remain unknown for 4,008 reached Access tasks; 24 have no original Access request. Contract identifiers alone do not establish parser equivalence. These interface discrepancies do not establish additional semantic failures.

Independent Answer on tasks without a workflow Answer. Figure 5 evaluates the corresponding tasks independently with correct inputs. Original workflow absence does not show that Answer would fail with sufficient information. This comparison does not assign a cause to the missing call or substitute an independent answer into the original workflow.

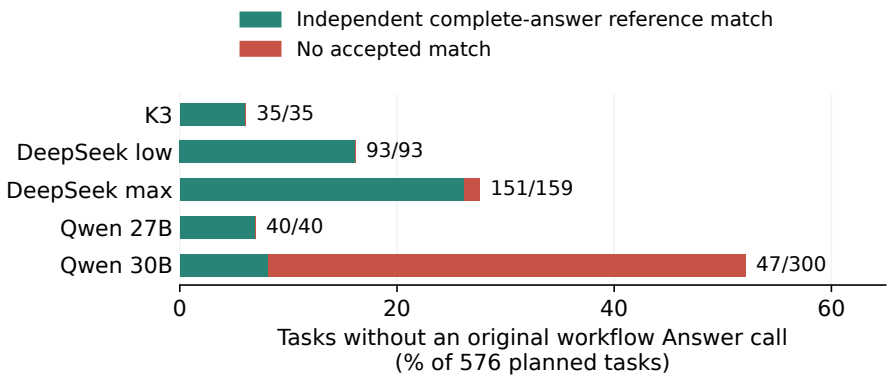


Figure 5: **Independent answering on tasks without an original workflow Answer call.** Labels give complete-answer reference matches/absent calls; bar widths use all 576 planned tasks. Original limits. These tasks differ from the readout-conditioned subset in Figure 3.

B.2 State size and knowledge usable after updates

Table 7 separates output size from knowledge usability. Sizes include only normally completed, nonempty State outputs; readout counts include every planned question. A small State is not necessarily a good State, and different output allowances prevent interpreting these ratios as a controlled comparison of compression quality.

K3 and DeepSeek low retain high usability after updates, whereas the Qwen3.8-27B actual-State stable readout decreases while reference-State readout remains high. This difference motivates inspecting the model's retained content. It does not directly measure forgetting: each score also depends on a reader call. DeepSeek max's reference-State readout also decreases, showing why actual readout changes cannot all be assigned to Update. Reference scores should not be mechanically subtracted.

Direct content checks are narrower. A fixed four-world, two-model sample checks 104/104 core commitments across 16 States, including eight additions and 48 unchanged commitments. It does not establish correctness of all revisions. The trace in Appendix C.4 shows that applying a notice can coexist with retaining an earlier error.

B.3 Family differences and conditional comparisons

Table 8 shows where aggregate function scores hide task differences. DeepSeek max obtains sufficient reference-State Evidence on only 31/144 purchasing tasks, versus 142/144 qualification tasks. Completion limits contribute strongly: across its actual and independent Access calls, 296/1,140 stop at length, including 210/285 purchasing calls. DeepSeek low has four length stops among 1,152 Access calls under the same ceilings. This is a configuration comparison, not evidence that more reasoning intrinsically reduces selection ability.

For Qwen3-30B-A3B, replay of reference-State requests executes 139/144 qualification requests but only five obtain sufficient Evidence; 142/144 service requests execute in replay and 46 are sufficient.

Table 7: State size and readout under original allowances. Panel A: medians over n normally completed nonempty States, out of 192 per configuration; ratios are computed per output before taking the median. Panel B: accepted State-only answers out of the header denominators. A/R denotes actual versus reference State. Stable counts are before → after Update. The panels describe different units; their association is not a length intervention.

A. Initial and updated State sizes

Configuration	n/192	Tokens	Source (%)	Reference ratio
K3	190	1206	10.70	1.477
DeepSeek low	192	1363.5	12.28	1.698
DeepSeek max	181	1660	15.40	2.044
Qwen3.8-27B	175	2045	19.11	2.523
Qwen3-30B-A3B	168	1319.5	14.75	1.729
Falcon-H1-7B	183	500	5.01	0.650
Llama-3.1-8B	140	742.5	7.94	0.940

B. Required-knowledge readout

Configuration	State	Initial /576	Updated /606	Revised /66	Added /30	Stable /510 each
K3	A	571	603	66	30	$506 \rightarrow 507$
	R	575	604	66	30	$509 \rightarrow 508$
DeepSeek low	A	563	592	62	29	$498 \rightarrow 501$
	R	570	596	63	30	$505 \rightarrow 503$
DeepSeek max	A	557	551	57	28	$494 \rightarrow 466$
	R	557	554	56	26	$493 \rightarrow 472$
Qwen3.8-27B	A	558	566	59	30	$494 \rightarrow 477$
	R	567	601	64	30	$503 \rightarrow 507$
Qwen3-30B-A3B	A	251	262	21	7	$230 \rightarrow 234$
	R	340	334	17	9	$325 \rightarrow 308$

No purchasing or assembly request is replay-executable: 143/144 purchasing requests exceed the public operation limit, and 143/144 assembly requests use disallowed record types. Thus interface compliance and incomplete selection both matter, but in different families.

Table 9 supplies uncertainty for the conditional comparisons in Figures 3 and 5. We resample initial worlds 10,000 times, stratified by family with 24 worlds per family, keeping each world's questions and rounds together. Intervals are pointwise 95% percentiles; resamples with a zero conditional denominator are omitted. They describe this world population, not generation variability. An all-success interval does not imply certainty outside the tested tasks.

B.4 SMALLER-MODEL FUNCTION TESTS

Independent tests provide the correct State and, for Answer, sufficient Evidence. Table 10 therefore compares functions without requiring a successful upstream workflow. Falcon-H1 and Llama also have actual workflows; the other four configurations do not. Missing workflow conditions are not scored as failures.

The independent results expose different limitations. Falcon-H1 answers 559/1,182 State-only questions but obtains sufficient Evidence on only 66/576 tasks. Qwen2.5-1.5B answers 422/1,182 State-only questions; all 45 of its executable requests concern qualification tasks and omit required provider–criterion combinations. None supplies sufficient Evidence. These are incomplete selections despite executable requests.

Llama's zero Recall matches instead comprise 612 invalid-syntax outputs, 379 invalid shapes, 175 semantic mismatches, and 16 length stops. Some noncompliant responses contain a correct answer object but fail to follow the requested interface. Interface failure alone does not establish inability to perform the semantic function.

Table 8: Access counts refer to saved-request replay. Family profiles at original allowances. Recall entries are matches/planned questions; Access columns count executable requests and sufficient selections separately, each out of 144 tasks with reference State. Actual Answer matches include all 144 planned tasks per family. Each family has 24 initial worlds.

		Recall matches		Reference Access /144		Actual Answer
Configuration	Family	Actual	Reference	Executable	Sufficient	Matches /144
K3	Purchasing	300/300	299/300	135	133	113/144
	Assembly	287/294	292/294	144	144	138/144
	Qualification	294/294	294/294	144	144	144/144
	Service	293/294	294/294	144	144	144/144
DeepSeek low	Purchasing	300/300	300/300	95	89	59/144
	Assembly	276/294	287/294	129	129	126/144
	Qualification	291/294	292/294	144	143	143/144
	Service	288/294	287/294	144	144	143/144
DeepSeek max	Purchasing	294/300	297/300	31	31	30/144
	Assembly	272/294	278/294	111	111	127/144
	Qualification	275/294	282/294	142	142	128/144
	Service	267/294	254/294	107	107	119/144
Qwen3.8-27B	Purchasing	267/300	300/300	130	130	102/144
	Assembly	294/294	293/294	144	144	143/144
	Qualification	288/294	294/294	144	144	141/144
	Service	275/294	281/294	144	144	132/144
Qwen3-30B-A3B	Purchasing	163/300	188/300	0	0	0/144
	Assembly	28/294	124/294	0	0	0/144
	Qualification	113/294	148/294	139	5	0/144
	Service	209/294	214/294	142	46	28/144

Table 9: Conditional rates behind Figures 5 and 3, with initial-world bootstrap intervals in percent. Left: independent Answer matches among tasks without an actual Answer call. Right: sufficient selection under public-rule replay among tasks with accepted target readout from identical actual State. These are different conditional task subsets, not semantic attribution rates.

	Independent Answer		Selection after readout	
Configuration	Matches / absent	95% interval	Sufficient / selected	95% interval
K3	35/35	100.0–100.0	185/192	93.8–98.4
DeepSeek low	93/93	100.0–100.0	158/182	83.3–90.2
DeepSeek max	151/159	91.3–98.1	119/170	64.5–75.3
Qwen3.8-27B	40/40	100.0–100.0	172/178	93.6–98.9
Qwen3-30B-A3B	47/300	10.4–21.5	11/56	11.1–29.3

With model-produced State, Falcon-H1 has 259/1,182 Recall matches and, in saved-request replay, 107/576 executable requests and 62/576 sufficient selections. Separately, 10/107 original Answer calls match. Llama has 11/1,182 Recall matches; replay finds three executable requests and no sufficient selection. None of its three original Answer calls matches. These conditions use model-produced State and Evidence, unlike the independent tests.

Llamba's input plus reserved output exceeds its configured window for 1,692/2,430 independent calls: all 576 Access and 96 Update calls, 588 Recall calls, and 432 Answer calls. Its 738 submitted calls comprise 594 Recall and 144 Answer calls, with no accepted matches, 430 length stops, and 308 normally terminated responses. No Access or Update capability score is available under this configuration; window exclusion is not a semantic failure.

Table 10: **Independent function tests for smaller models.** Recall and Answer count accepted reference matches, including the required output format. Access counts refer to saved-request replay, separating executability from Evidence sufficiency with reference State. Update counts normal State returns, not content correctness. Length stops span all 2,430 independent calls in each row; they are included in planned denominators and are not semantic-error counts. Llamba's partial input-window coverage is reported separately below.

	Recall	Access		Answer	Update	Length
Model	/1182	Executable /576	Sufficient /576	/576	/96	stops
Llama-3.1-8B	0	1	0	7	96	83
Falcon-H1-7B	559	116	66	41	86	170
Qwen2.5-1.5B	422	45	0	8	96	15
Falcon3-Mamba-7B	113	0	0	13	60	435
Zamba2-7B	67	0	0	14	84	484

C Interpreting Selection and Answering Failures

C.1 Checking the information actually supplied

We judge answering support against the public sources and task requirements, not from whether the final answer matches. For each inspected case, saved calls identify the configuration, world, round, and question. Deterministic checks verify the State supplied to the call and the text materialized from its request. An auditing LLM agent then inspects the complete State and Evidence for the applicable relationships, rules, authority, scope, and effective time, including contradictory or superseded statements. Record comparisons and calculations check the source facts against these judgments.

Inputs are sufficient when their verified relevant contents are correct, applicable, and jointly determine the answer. Extra facts retained in State count toward support; missing retrieved facts do not imply insufficient joint inputs if State supplies them. A missing or incorrect requirement makes the inputs insufficient. Without a completed support check, they remain unverified. Neither Recall success nor Evidence sufficiency with reference State replaces this inspection.

The original mismatch and readout–selection reviews use a single auditing agent per case, not blinded human annotation. Exact reviewer model versions were not recorded. Scripts reproduce record comparisons and calculations, but not the free-text semantic judgments. Reviewers could see outcomes; this procedure has no measured inter-annotator agreement. The separately reported expanded-condition cases received two agent reviews. The evidence supplement preserves the full State, original requests, returned Evidence, source documents, and existing judgments for the four selection failures under correct task-relevant State and the two answering contrasts below. A companion record index locates each task, State, request, and returned record; sidecars recover two originally omitted Access inputs. Its verifier checks the qualification example's twelve required versus nine returned scores and recomputes the answering contrasts. These checks do not certify free-text State semantics or estimate annotation reliability.

C.2 Coverage and complete-answer mismatches

Table 11 reports the full scope of original-input inspection: 113 of 2,358 nonempty Answers, comprising 75 sufficient and 38 insufficient input pairs. The other 2,245 are unverified. This deduplicated set includes the 39-case mismatch cohort and the 23-case readout–selection cohort below. It is selected evidence, not a benchmark-wide error partition or a comparable use-error rate across models.

The complete cohort of 39 normally completed, format-valid mismatches in K3, DeepSeek low/max, and Qwen3.8-27B spans 25 worlds:

Table 11: **Actual-input attribution coverage.** Seven original-generation workflows, each with 576 planned tasks. Sufficiency concerns independently checked task-relevant actual State plus Evidence. Unverified denotes inputs without a sufficiency judgment, not semantic failure. Inspections are selected.

	Nonempty	Verified joint inputs		Unverified
Configuration	Answers	Sufficient	Insufficient	inputs
K3	541	27	2	512
DeepSeek low	483	15	0	468
DeepSeek max	412	26	2	384
Qwen3.8-27B	536	7	11	518
Qwen3-30B-A3B	276	0	23	253
Falcon-H1-7B	107	0	0	107
Llama-3.1-8B	3	0	0	3
Total	2358	75	38	2245

- **Insufficient inputs (15).** Eleven involve State truncated at a generation limit, including two Updates inheriting truncated Builds; four contain wrong bindings or policy values. Five also lack required Evidence. Ten appropriately abstain under the supplied information. These are task mismatches, not evidence-use errors.
- **Wrong facts or results with sufficient inputs (5).** All occur in Qwen3.8-27B: one qualification answer substitutes a score of eight for three; two service answers use another service's lead time; one uses another service's capacity; one abstains despite an available qualifying offer. The required information is present in each case.
- **Correct task results, intermediate-field mismatches (19).** Eighteen assembly answers report post-startup quantities where the reference reports per-component kit capacities, while finished kits, bundles, and remainders are correct. One qualification answer preserves scores, waivers, and decisions but reports a margin of -12 instead of $1 - 7 = -6$.

All 19 remain nonmatches under complete-answer scoring. The assembly instructions prescribe startup subtraction, floor division, and minimum production, but do not separately state the capacity field's units; the qualification instruction requests a "real margin" without its formula. These omissions leave uncertainty about the intended field interpretation. We therefore report the scoring discrepancy without classifying these cases as established semantic use errors or claiming that the models necessarily interpreted the requirements correctly.

C.3 Complete answers versus task results

Table 12 applies one field map to every Answer in all 11 original configurations, including successful outputs and both actual/reference conditions. Both scores require normal completion and the public format. Complete-answer success checks every field. The task-result score omits value matching for three intermediate calculations: per-component kit capacities in assembly, per-component costs in assembly quotation, and qualification margins. It still requires all entity identities, reported facts, policies, qualification decisions, and final operational results. All fields count in purchasing and service tasks, including prices, reserve gaps, packing, capacity, lead time, and ready time. Intermediate fields remain mandatory; this additional measure neither accepts missing fields nor repairs syntax. The supplementary clarification states the retained interpreter conventions: component capacity counts whole kits after startup loss; quotation cost is unit price times units per kit; qualification margin is score minus the minimum, including waived criteria. It preserves the evaluated prompts and complete-answer scorer and freezes the additional task-result field map. Its offline verifier checks all 576 reference tasks. No model received this clarification: it neither changes reported scores nor resolves how models interpreted the original wording.

Across 10,368 planned Answer slots, 90 outputs have correct task results but mismatching intermediate values: 85 assembly and five qualification, comprising 27 actual and 63 independent responses. No

Table 12: **Complete answers versus correct task results.** Original generation, 576 planned questions per evaluated condition. Both columns require normal completion and the existing valid output format. "Results" additionally ignores only intermediate arithmetic values identified in the field map; all entity identities, reported source facts, and final results must match. Complete-answer counts are unchanged. A dash means the workflow was not evaluated. Counts over planned tasks do not turn unavailable calls into demonstrated semantic failures.

	Actual inputs		Reference inputs	
Configuration	Complete	Results	Complete	Results
K3	539	539	576	576
DeepSeek low	471	482	562	573
DeepSeek max	404	410	525	536
Qwen3.8-27B (thinking)	518	520	569	575
Qwen3-30B-A3B	28	36	102	118
Falcon-H1-7B	10	10	41	58
Llama-3.1-8B	0	0	7	9
Falcon3-Mamba-7B	–	–	13	13
Llamba-8B	–	–	0	0
Qwen2.5-1.5B	–	–	8	8
Zamba2-7B	–	–	14	14

quotation-cost response changes. These repeated task/configuration observations are not independent samples, and correct task results do not certify input support.

C.4 Readable knowledge and source selection

We inspect all 23 Qwen3-30B-A3B tasks with accepted updated-target Recall, identical State at Recall and Access, replay-executable requests, and Evidence insufficient with reference State. They span 16 worlds. Nineteen requests choose a wrong department, two omit a supplier, and two have an additional invented role–purpose restriction in State. Seventeen of the 21 States with correct operational content have inaccurate or incomplete update metadata. Only four have fully correct task-relevant content, scope, and provenance. All 23 joint inputs lack required source facts, with no compensation from extra facts in State.

Thus the whole cohort supports a readout–selection discrepancy, while only four cases support the stronger explanation of failed selection despite correct task-relevant State. Among the 56 tasks with correct actual-State readout and identical State at readout and Access, additionally requiring correct reference-State readout leaves 38 tasks, with 26 executed requests and nine sufficient selections. These readout results still do not certify actual State content.

Two distinct places to intervene. Table 13 contrasts an incorrect retained binding with a selection omission under correct State. In the assembly case, an unrelated update changes another production rule and leaves the wrong binding intact. Access follows that binding, and Answer calculates the cost consistently from the wrong line's genuine prices. In the qualification case, three operations can cover all four suppliers by requesting one criterion per operation. Omitting a supplier is not forced by the request limit. The two examples require different repairs.

A contrasting service case (Qwen3.8-27B, world 44402, round 1, question 3) has correct State and sufficient Evidence. Grove is the first qualifying service, with lead time one and handling time four. Answer instead uses Dover's lead time eleven, reporting ready time fifteen rather than five. This is a use error under sufficient inputs.

Correct State can lead to different requests. In two purchasing quotations, K3's actual and reference States both contain the required role-to-clinic mapping. With actual State, it selects item lists, contracts, and prices (83, 26, 81 in one task; 20, 21, 38 in the other). With reference State, it rereads the mapping and item list but obtains no prices. Each case has six successful independent

Table 13: Selected traces locating different defects. The assembly case is K3, world 45414, quotation task; the qualification case is Qwen3-30B-A3B, world 46421, round 2 questions 1–2. These inspected cases are not error rates.

	Wrong retained binding	Correct State, missing records
Task requirement	Emergency dispatcher → North berth → South line.	Reserve → South panel; evaluate four suppliers on Cold-chain, Insurance, and Licensing.
Actual State	Maps the dispatcher to North instead of South.	Correct binding, threshold seven, Insurance waiver, and notice scope.
Request and Evidence	Selects North-line prices 1, 13, 1 instead of South-line prices 9, 14, 1.	Retrieves nine scores for Alder, Birch, and Cedar; omits Dover's three required scores.
Consequence	Wrong-line bundle cost: $4(1 + 13 + 1) = 60$. Required: $4(9 + 14 + 1) = 96$.	Neither State nor Evidence supplies Dover's three scores; the waived criterion also requires a reported score.

Recall answers, which are not supplied to Access. The State representations differ; these pairs do not isolate a representation-format effect.

Sensitivity to repeated selection. Eleven selected divergent tasks have five repeats per State condition (110 calls). For the wrong-line case, actual State produces sufficient selection on 0/5 repeats versus 5/5 with reference State: four actual requests execute on the wrong line and one violates the schema. The price-omission cases are less stable, with actual/reference sufficiency 4/5 versus 3/5 and 4/5 versus 4/5. Two matched control tasks across three configurations have 58/60 sufficient selections; two DeepSeek max reference calls stop at length. Settings are matched within each comparison; K3 uses temperature 1.0 and streaming, and one DeepSeek case uses a fourfold allowance. These selected repeats test persistence, not population stability.

C.5 Correcting information supplied to Answer

We select 24 tasks from 14 worlds based on observed outcomes: four with State and Evidence defects, four with only Evidence defects, twelve with observed sufficient-input errors, and four successful controls. They comprise ten purchasing, five assembly, three qualification, and six service tasks. The deliberate selection tests repair responses, not prevalence.

For the eight information-defect tasks, we cross actual/corrected State with actual/complete Evidence at Answer. State correction replaces only the relevant binding and policy; complete Evidence supplies the required current facts. Access is not rerun. All four conditions use one neutral record representation with type, subject, and facts, excluding addresses and provenance. A baseline is generated in that same representation, so changes are not compared against differently formatted answers. In four Evidence-only cases the two States are identical; identical conditions reuse one response. The other 16 tasks receive no reminder, a generic reminder, or a fixed family-rule reminder without target facts. This gives 80 task–condition combinations but 72 distinct calls.

DeepSeek V4 Pro uses provider-default temperature with thinking disabled; output allowances are 512/768/1,536/512 for purchasing/assembly/qualification/ service. A second configuration uses maximum thinking with 16,384 additional tokens per call. The same 72 input messages are used, with 56 sufficient and 16 insufficient input pairs in each configuration (Table 15).

The matched-format baselines have no reference-answer matches in either four-task group under either configuration. All baseline and corrected responses complete normally. Without thinking, baselines include two valid abstentions and one format violation; with maximum thinking, all eight abstain. These are nonmatches under insufficient inputs, not semantic use errors. The supplementary addendum records all eight paired tasks under both settings, including response reuse.

Table 14: **Does repairing the inputs also repair the answer?** DeepSeek, thinking disabled; eight selected tasks, separate from the 39 inspected responses. The four middle columns count sufficient inputs; the last gives reference-answer matches in the matched-format baseline and after correction. Baseline nonmatches include abstentions and one format violation under insufficient inputs.

	Sufficient inputs after correction				Answer matches
Defect	None	State	Evidence	Both	baseline → corrected
State and Evidence	0/4	0/4	0/4	4/4	$0/4 \rightarrow 3/4$
Evidence only	0/4	0/4	4/4	4/4	$0/4 \rightarrow 4/4$

Table 15: **Answering identical inputs with two resource configurations.** DeepSeek V4 Pro on 24 selected tasks from 14 worlds: 72 distinct prompts per configuration, including repeated conditions for the same tasks. More resources combines maximum thinking with 16,384 additional output tokens per call. These are selected comparisons, not population recovery rates.

Inputs	Outcome	Thinking disabled	More resources
Sufficient (56 calls)	Correct	46/56	54/56
	Semantic error	10/56	1/56
	Length stop	0/56	1/56
Insufficient (16 calls)	Valid abstention	5/16	15/16
	Other response	10/16	1/16
	Format error	1/16	0/16

After complete Evidence is supplied, all four Evidence-only answers match in both configurations. The four joint-defect cases need both corrections for sufficient inputs; three answers then match without thinking and all four with more resources (Table 14). One answer happens to match with wrong State and corrected Evidence; it is not counted as successful use of sufficient, correct inputs. Among 55 sufficient-input pairs with final answers, 44 are correct under both settings, ten only with more resources, and one only without: 45/55 versus 54/55. The remaining higher-resource call stops at length. Reminders show no consistent advantage. These results distinguish repairing information from using it correctly; they do not isolate reasoning effort from allowance or estimate a benchmark-wide repair rate.

D NATURAL-TEXT TASKS

This adaptation expresses existing natural-text relationships in the same task format while retaining original source sentences. We assemble six groups from the union of HotpotQA distractor development paragraphs (Yang et al., 2018), each with four films and four director biographies. Groups are fixed before the new questions, though the source pool was already curated for HotpotQA's own questions. Build retains every film–director relationship before seeing questions; new questions ask for each film director's birth year. One malformed source date is excluded without replacement, leaving 23 questions. Its binding remains in State. This is not an evaluation on the original HotpotQA question set.

Sources contain 557–909 tokens, reference States 60–64, with the counting method in Appendix A.2. In the actual workflow, the model prepares one directory per group, selects a person and attribute, and answers from its State and the biography supplied by an exact-title resolver. No reference State or request repairs that trajectory. Both models produce six semantically correct States, 23 correct person selections, and 23 source-supported correct Answers (Table 16).

Controls and scoring. BM25 (Robertson & Zaragoza, 2009) searches titles and bodies with $k_1 = 1.2$, $b = 0.75$ and returns two whole paragraphs. State-guided retrieval appends the director's name to the question and supplies State to Answer. The same-Evidence control adds State only after question-

Table 16: Natural-text workflows and controls on the same 23 questions. Answer columns count source-supported correct content, separately from format compliance. Biography counts measure retrieval; they are not Answer scores. State-guided controls use reference State before retrieval and at Answer. Dashes denote inapplicable or untested conditions.

Condition	Biography /23	DeepSeek /23	K3 /23
Model-produced workflow	23	23	23
Question-only BM25, two paragraphs	4	4	4
Same Evidence + reference State	4	4	4
State-guided BM25, two paragraphs	21	21	20
Question-only dense, two paragraphs	11	11	—
State-guided dense, two paragraphs	23	23	—
Reference State + correct biography	23	23	23
Full source group	23	23	23

only retrieval, holding its paragraphs fixed. The dense control uses `BAAI/bge-small-en-v1.5` (Xiao et al., 2024), CLS pooling, cosine similarity, and its query prefix. Paragraph counts, not token budgets, are matched. Dense controls reuse existing responses when the selected paragraph set is identical; two guided dense responses retain the recorded paragraph order rather than the new ranking order. Their scores describe those recorded inputs, not fresh order-matched generations. Correct years alone are insufficient: checks verify cited titles and sentence indices, the retained film–director relationship, and support in the actual answering inputs.

Adding State after the same inadequate BM25 selection leaves four supported answers; using it to form the query yields 21 for DeepSeek and 20 for K3. Both retrievers select more target biographies with State in these two-paragraph controls. This explains an effect on selection within this adaptation, not a necessity of prepared State for natural-text retrieval. An iterative retriever could follow the film–director link without it.

Generation and scope. DeepSeek uses thinking disabled and default temperature, with Build/Access/ Answer allowances of 512/128/256 tokens. Its model workflow and six answering controls contain 190 calls, all normally completed. Bare-JSON compliance in the workflow is 5/6 Builds, 21/23 requests, and 21/23 Answers; eight responses include a single code fence. Their content scores are not strict interface-compliance scores. K3 uses temperature 1.0, default thinking, streaming, and the same limits. Of its 190 calls, 186 terminate normally; four stop at length (one guided Answer, one same-Evidence Answer, one closed-book Answer, and one workflow Access). The last still yields a parseable request followed by a correct Answer.

Closed-book year matches are 1/23 for DeepSeek and 22/23 for K3, so correct years do not establish dependence on the supplied sources. Two questions also have single-biography shortcuts, and two directors recur across groups. Questions share sources and State; this short, one-template sample has no Update and establishes neither natural long-context performance nor synthetic/natural difficulty equivalence. Source titles, original sentence indices, attribution, and CC BY-SA 4.0 licensing are retained.

E Sensitivity and Resource Controls

E.1 Larger generation allowances

These selected tests are separate from original-generation results. They increase allowances fourfold while keeping the model's thinking setting fixed. Selection follows observed length stops, so the results describe those calls rather than the full task population. We distinguish unchanged messages from trajectories in which regenerated State also changes the downstream input; unchanged outputs from unselected tasks are not added to either set.

Table 17 gives the Qwen and smaller-model comparisons. For K3 and DeepSeek, 425 selected fixed-input calls return normally with larger allowances. A further Access call has different State and is excluded from that fixed-input count. Seventy selected Build/Update calls (seven K3, 63 DeepSeek max) also return, followed by 168 executed calls out of 170 downstream calls on the resulting States. Return counts measure completion, not State correctness or answering quality.

Table 17: Selected fourfold-allowance comparisons. Panel A uses Qwen3.8-27B: normal completion is original $\rightarrow$ larger allowance, and Access outcomes use the final public rules and reference-State sufficiency. Panel B uses fixed reference inputs for smaller models. Its rows have different selected operation mixtures; they are not a ranking. State returns do not establish content correctness.

A. Qwen3.8-27B: separate input conditions

Input condition	Operation	Calls / worlds	Normal	Outcome
Identical	Build	4 / 4	$0 \rightarrow 4$	4 State returns
	Update	63 / 58	$0 \rightarrow 63$	63 State returns
	Access	22 / 15	$0 \rightarrow 22$	$0 \rightarrow 22$ sufficient
Changed State	Update	3 / 3	$3 \rightarrow 3$	3 State returns
	Recall	101 / 13	$99 \rightarrow 101$	$77 \rightarrow 101$ matches
	Access	48 / 13	$46 \rightarrow 48$	48 sufficient at larger limit
	Answer	40 / 12	$40 \rightarrow 40$	$32 \rightarrow 39$ matches

B. Smaller models: identical input, larger allowance

Model	Normal	Length stop	Recall	Access sufficient	Answer
Falcon-H1-7B	74/170	96/170	30/91	2/27	3/42
Llama-3.1-8B	4/4	0/4	0/1	0/3	—
Zamba2-7B	12/28	16/28	0/20	—	0/8
Qwen2.5-1.5B	0/4	4/4	0/1	0/3	—

Qwen's 89 identical-input pairs all stop at length under original allowances. With more allowance, all 22 Access requests execute and supply sufficient Evidence. Its other 192 paired calls receive different State; one Answer also receives different Evidence. Therefore their changes cannot be assigned to a larger local allowance alone. Temperature, top-p, top-k, and thinking remain unchanged, but sampled generations lack paired random seeds, so even fixed-input transitions do not prove causal necessity.

The Qwen test has 348 planned calls over 70 worlds, including 67 without a valid original/follow-up pair: one Update, 24 Recall, 12 Access, and 30 Answer slots. Of these, 64 return and three Answers are absent. Across the selected test, actual and reference Access supply sufficient Evidence on 70/70 and 12/12 tasks; recorded actual Answers match on 65/67 returns, or 65/70 planned tasks. A scored request supplies no missing Answer output, and an answer from another State trajectory is not substituted.

Smaller-model comparisons cover 206 selected calls in 82 Falcon-H1, four Llama, 15 Zamba2, and three Qwen2.5 worlds (within-model counts). Falcon's calls include ten Updates, five normal. Among its 69 normal Recall/Access/Answer outputs, 52 satisfy the interface, 12 have invalid shape, and five violate the public Access schema. Llama's four normal outputs fail syntax; Zamba2's twelve fail syntax or shape. More allowance therefore leaves substantial interface failures as well as length stops.

Two inspected expanded-condition answers. The two Qwen mismatches above are distinct from the original 39-case cohort. Both have verified sufficient actual inputs. A qualification response has correct scores, waivers, and decisions but a nonmatching intermediate margin. A service response chooses the correct first qualifying offer but uses a later offer's capacity and time: it reports ten seats and ready time 16 instead of eight and 13. The latter is a fact/use error; the former is a complete-answer failure confined to an intermediate field.

E.2 Preparation instructions and retained facts

Eight fixed K3 worlds, two per family, are selected by sorted identifier among normally completed Builds. We compare full retention instructions, removing only the instruction discouraging current source records, and also removing the required-content specification. All three remain query-blind and retain metadata requirements. K3 uses temperature 1.0, default thinking, and the same family allowances in every condition; this control has no Update (Table 18).

Table 18: Preparation instructions on the same eight worlds, 48 State-only questions, and 24 source-dependent tasks per condition. State lengths include all eight outputs, including empty or truncated text. Retained facts count none/part/all of a checked task's source-record support, excluding generic names and policy vocabulary. Joint sufficiency is checked only for called Answers.

	Full instructions	Allow records	Also omit content list
Normal / nonempty / truncated Builds	7 / 8 / 1	7 / 7 / 1	1 / 8 / 7
Median State tokens	818.5	1,163	3,619.5
Task facts: none / part / all	24 / 0 / 0	21 / 0 / 3	7 / 5 / 12
Recall matches /48	45	42	11
Sufficient Evidence /24	23	21	4
Complete-answer matches /24	21	21	2
Called Answers /24	24	21	13
Sufficient / insufficient joint inputs	21 / 3	21 / 0	1 / 12

All 72 task–instruction combinations and 393 task–record entries receive exact/normalized matching, structured fact comparison, and content review. The 58 called Answers receive the inspected State unchanged. Extra facts are credited toward support. Removing only record-omission guidance yields the same 21/24 correct Answers and longer median State; one world retains all three tasks' source facts, duplicating sufficient returned Evidence. Without the content list, five called tasks obtain some missing facts from State but still lack required rules or bindings. None gains sufficient joint inputs from that supplement. Seven of eight Builds then stop at length. This does not show that longer State causes failure or that content guidance is universally necessary. The interface requires a nonempty Access request; State-only answering and retrieval skipping are not tested.

E.3 Selected Evidence versus full sources

We use 48 matched tasks from eight fixed worlds, two per family, with two rounds and three questions per round. DeepSeek answers from full sources, reference State plus full sources, or reference State plus program-selected Evidence. The selector translates State and public question fields into typed family requests, without reference answers; it is not a model Access test. All three conditions supply sufficient information.

The thinking-disabled configuration uses default temperature and allowances 512/768/1,536/512 by family. Maximum thinking adds 16,384 output tokens; it changes effort and allowance together. Table 19 keeps incomplete outputs separate from incorrect final answers.

With limited resources, selected Evidence gives 23 correct answers versus two from full sources and six from State plus full sources. With more resources, 41 tasks have scorable answers in all three conditions: 40/41, 40/41, and 41/41 are correct, respectively. Raw totals such as 45 versus 44 should not be read as a full-source advantage: their unscored tasks differ. Full sources largely catch up within this matched sample.

Two additional controls use question-only BM25 (Robertson & Zaragoza, 2009) ($k_1 = 1.5$, $b = 0.75$, 512 tokens per passage and 2,048 total Evidence tokens), with or without reference State added after retrieval. They supply sufficient joint inputs on 1/48 and 4/48 tasks; zero and one of those answers are correct. They are acquisition controls, not comparisons of reasoning on equally adequate inputs.

Selected Evidence reduces answering-stage prompt input: with thinking disabled, totals are 107,203 tokens versus 841,780 for full sources and 896,101 for State plus full sources. Completion totals

Table 19: Answering 48 matched tasks under two resource configurations. All listed inputs are sufficient. Thinking-disabled outputs all complete in valid format. More resources means maximum thinking plus a larger output allowance. Its four outcome columns sum to 48 for each input condition.

	Thinking disabled	More resources			
Input	Correct /48	Correct	Incorrect	Length stop	Incomplete
Full sources	2	45	1	1	1
State + full sources	6	46	1	0	1
State + selected Evidence	23	44	0	3	1

are 6,997, 6,932, and 6,389, respectively. Among 45 tasks with recorded usage in all three input conditions with more resources, selected Evidence uses 104,143 prompt and 354,475 completion tokens, versus 785,294/235,245 and 835,979/263,374 for the full-source conditions. Completion includes reasoning. Lower input volume therefore does not imply lower completion cost. Preparation, indexing, and retrieval costs are excluded, so this is not an end-to-end efficiency result.

F Detailed Comparison with Prior Work

Table 20 compares observable behavior and research questions; we do not report performance comparisons with these methods. SEA-Boundary specifies how retained knowledge guides later source requests, and checks whether actual State and Evidence jointly support an inspected answer.

Table 20: Related work and the corresponding focus of SEA-Boundary. Prior diagnostics already inspect multiple stages; our focus is the connection between pre-question preparation and later source reading.

Work	What it examines	SEA-Boundary focus
EvalMem (Liu et al., 2026d)	Stored facts, native retrieval, and oracle-context answering; generated state changes.	How retained knowledge determines additional source needs; actual joint-input support.
AMemGym (Cheng et al., 2026)	Structured user states; memory writing, reading, and use; correct-state control.	Retained rules and mappings used to request and interpret source facts.
InMind (Li et al., 2026)	Direct recall versus implicit application; controls supplying target memory.	Applying retained knowledge during source selection as well as answering.
ReadAgent; GAM (Lee et al., 2024; Yan et al., 2025)	Gists or memos, original pages, and source-reading actions.	Task-defined checks of what preparation preserves and later reading supplies.
MemR^3 (Du et al., 2025)	Evidence, unresolved needs, query revision, and stopping in iterative retrieval.	Inspectable one-shot requests and their returned information.
WorldMemArena; StreamMemBench (Liu et al., 2026a;b)	Evolving memory, later tasks, and checks of additions and revisions.	Explicit updates and their effects on readout, source requests, and answers.
RAGChecker; WhenLoss; D^{2}ACCI (Ru et al., 2024; Yu et al., 2026; Liu et al., 2026c)	Fact coverage and use; fixed-reader input comparisons; process records.	Task requirements connecting State, requests, and returned Evidence.
Sufficient Context; RECON (Joren et al., 2025; Arya, 2026)	Context sufficiency; generated proof dependencies and full-coverage retrieval checks.	Whether actual State and Evidence jointly determine the required answer.
Hindsight Memory-PRM (Jia et al., 2026)	Answers after memory-entry deletion and renewed retrieval.	Derivability under declared rules; behavioral responses to input corrections.

Other diagnostics also go beyond pooled accuracy. MemTrace follows individual facts across time and evidence conditions (Long et al., 2026); UTILMEM tests integration of distributed conversational evidence (Qing et al., 2026). These studies motivate distinguishing recall, selection, and use. SEA-Boundary makes the requirements connecting them explicit in each generated task; generation or stage-wise scores alone are not the distinction.