MEMORYLOD: SOURCE-GROUNDED NAVIGATION OVER MULTIRESOLUTION MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon language agents must retrieve useful details from histories that grow beyond what they can read at once. Compact summaries can lose information needed by later questions, while reading entire histories consumes context and computation. We present MEMORYLOD, a memory framework that organizes source files into a navigable graph at multiple levels of detail. Inspired by level of detail in computer graphics, the graph combines coarse views for orientation with fine-grained access to source content. An agent explores a bounded view of this graph, follows local and cross-resolution links, and opens sources to gather evidence for its question. The graph is constructed independently of the question, while navigation and stopping are guided by the evidence the question requires. Current source files remain authoritative: derived representations guide retrieval, and only opened source evidence is used to support answers. Our primary evaluation uses BEAM, LongMemEval-V2, and AMA-Bench, covering conversational memory, multimodal interaction histories, and agent-history memory. The evaluation measures answer quality and evidence provenance, with diagnostic analyses that distinguish failures to find relevant information from failures to use it.

1 INTRODUCTION

Long-horizon agents need to recover specific facts from histories too large to inspect at once. A compact summary helps the agent find its way, but can omit the fact a later question needs. Reading every source in detail consumes attention and context before the agent knows where to look. Performance also depends on where relevant information appears within a long context (Liu et al., 2024). A memory system therefore needs to provide an overview, let the agent inspect a relevant region, and keep other directions available.

Level of detail as an observation policy. A similar problem arises in computer graphics, where *level of detail* (LOD) allocates a finite geometry budget across a scene. Nearby objects receive detailed meshes, while distant objects use fewer polygons. This leaves capacity to display many distant objects alongside a few detailed nearby ones (Clark, 1976; Hoppe, 1996).

We present MEMORYLOD, a source-grounded memory system that treats retrieval as navigation through an automatically rendered, mixed-resolution scene. The current node sets the scene's focus: detailed nearby source previews appear alongside coarse routing summaries of distant areas. The agent chooses where to visit and which sources to open; movement automatically reallocates detail. Routing summaries guide navigation, and only opened source evidence can support the answer.

Figure 1 illustrates this allocation: coarse routing summaries represent many distant areas, intermediate routing summaries distinguish groups, and nearby source previews receive finer detail, all within one scene token budget. The current node's location in a stable cover hierarchy determines proximity. The question selects informative text within visible routing summaries, while the hierarchy preserves the available directions. A coarse direction stays represented even when its wording has little overlap with the question.

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.

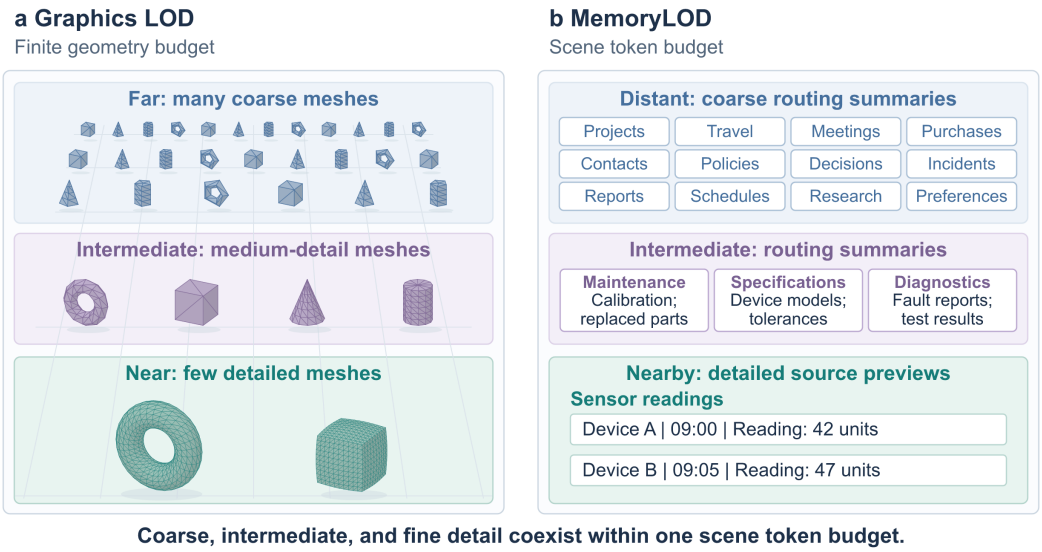


Figure 1: Three simultaneous levels of detail. Left: many distant objects use coarse meshes, intermediate objects retain more shape, and a few nearby objects receive fine meshes within a finite geometry budget. Right: MemoryLOD allocates its scene token budget to coarse and intermediate routing summaries and detailed nearby source previews. Matching colors show the correspondence. The three levels, content, and object counts are illustrative. Previews guide navigation; subsequent source openings supply evidence for answering.

MemoryLOD combines hierarchical organization and graph navigation by automatically reallocating detail while preserving structural coverage. It builds on hierarchical retrieval, active traversal, and file-backed memory (Sarthi et al., 2024; Edge et al., 2025; Dai et al., 2026; Zhang et al., 2026; Ji et al., 2026; Zhou et al., 2026), connecting routing summaries to opened source evidence through persistent source locators. *Structural coverage* is the fraction of indexed source units represented in the scene. Full structural coverage means every source unit is represented.

Workspace files can also provide persistent memory for long-running agents. A code repository carries context across tasks in its source files, documentation, and recorded decisions. MemoryLOD could help an agent navigate such a workspace through coarse views of modules and close inspection of relevant files as the task changes. Agent harnesses that store session histories as JSON records alongside pasted content, user-provided images, and tool-generated images offer another possible application. In both cases, the existing files would remain authoritative and the derived LOD representations would provide a rebuildable navigation layer. Repository and session memory remain topics for future study; the public benchmarks here assess agent memory mainly through question answering.

Our contributions are (i) a scene algorithm that preserves structural coverage and allocates resolution around the current node; (ii) source-grounded routing digests and source openings connected by a small action interface and bounded answer review; and (iii) an evaluation across conversational, multimodal, and agent-history memory, with exact source and execution replay.

2 Related Work

Retrieval-augmented generation supplies external information at inference time (Lewis et al., 2020). Hierarchical methods organize representations at several granularities: RAPTOR builds recursive summaries, and MemTree organizes memory in a growing hierarchy (Sarthi et al., 2024; Rezazadeh et al., 2025). GraphRAG, HiRAG, LeanRAG, and SLoD provide graph or multiscale access to related information (Edge et al., 2025; Huang et al., 2025; Zhang et al., 2026; Izgorodin, 2026). HippoRAG and HippoRAG 2 combine graph-based memory with Personalized PageRank retrieval (Gutiérrez et al., 2024; 2025). These methods provide compact views of large collections.

MemoryLOD specifies how resolutions coexist in one bounded scene and how that scene changes after a visit.

Iterative retrieval lets previously retrieved information inform later access. MemWalker (Chen et al., 2023) navigates a hierarchy of text summaries, and GraphReader (Li et al., 2024) explores a graph of long-document content. ReadAgent (Lee et al., 2024) uses episodic gist memories to guide lookups of original passages. IRCoT, NaviRAG, and MRAgent study reasoning interleaved with retrieval or explicit navigation (Trivedi et al., 2023; Dai et al., 2026; Ji et al., 2026). NaviRAG maps vector-retrieved chunks to candidate semantic subtrees and traverses them from coarse to fine, choosing whether to absorb a node's summary or expand its children. Its answer context combines initial chunks, selected summaries, and raw text.

SLoD controls granularity through a continuous diffusion scale: heat-kernel weights define a Fréchet aggregate of graph embeddings, and BoundaryScan identifies abstraction boundaries along that scale (Izgorodin, 2026). Its proposed retrieval interface selects a scale and accesses an aggregate or cluster at that scale. MemoryLOD renders a hierarchy cut across the indexed memory in each scene, exposing source previews near the focus alongside coarse alternative directions. A `visit` relocates detail automatically; `open` acquires opened source evidence, and `stop` requests completion.

Self-Refine (Madaan et al., 2023) iteratively revises outputs using generated feedback, while Self-RAG (Asai et al., 2024) integrates adaptive retrieval and self-reflection into generation.

MemGPT (Packer et al., 2024) manages information across memory tiers; the Letta framework extends this approach to agent memory (Letta, 2024). A-MEM (Xu et al., 2025) dynamically links structured memory notes and updates their contextual representations as new memories arrive.

File-backed systems expose memory through files and file tools, including AgentRunbook-C and Filesystem-Based Memory (Wu et al., 2026; Zhou et al., 2026); Semantic File Systems established derived access over ordinary files much earlier (Gifford et al., 1991). MemoryLOD retains source files as the authority and uses a rebuildable graph to access them. Its automatic scene composition brings these ideas together: routing summaries identify promising directions, and source openings supply evidence for answering.

3 Problem and Design Goals

Given a question q and a collection $\mathcal{F}$ of conversations, documents, or multimodal interaction histories, the task is to acquire supporting evidence and answer under a fixed model context capacity and a finite model-call budget. Relevant facts may span records whose interpretation depends on speaker, time, or surrounding events. MemoryLOD separates compressed representations for navigation from authoritative source content for answering, guided by four goals.

Preserve orientation while allocating detail. Expose detail near the agent's focus and retain coarse views of alternative directions. Movement should adjust this allocation automatically within the scene token budget.

Preserve context when acquiring evidence. Open coherent passages with their speaker, temporal, and structural context, while reserving evidence characters and image slots for complementary evidence. Multimodal states retain their associated images.

Distinguish inspection from support. Track both inspected source content and unresolved question requirements. A comparison requires evidence for both sides at compatible scopes and times. Tracking these requirements directs retrieval toward missing evidence.

Allow correction within explicit resource limits. Allow synthesis and review to redirect navigation toward an evidence gap that matters to the answer, within the original model-call and evidence character budgets. Unresolved central facts call for qualification or abstention.

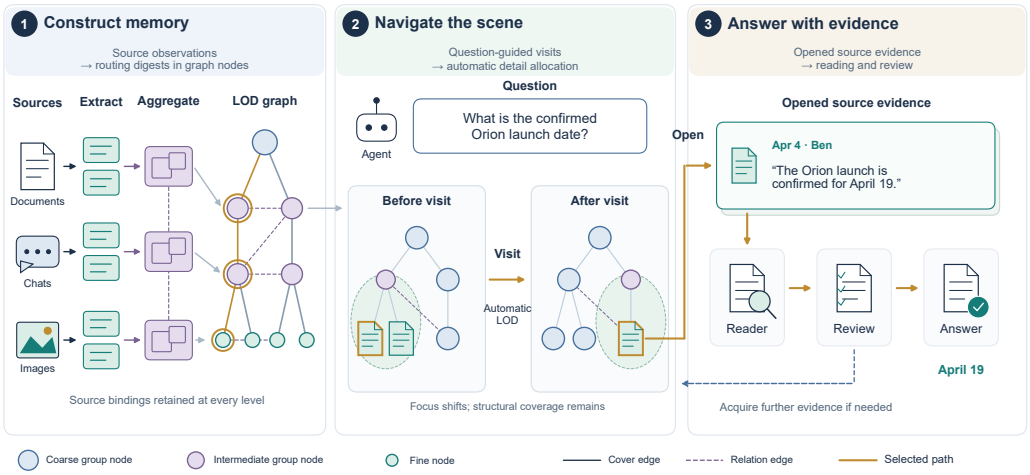


Figure 2: MemoryLOD architecture, from construction to evidence-grounded answering. **1:** Source observations extracted from heterogeneous sources retain their source bindings. Routing digests aggregate these observations at fine, intermediate, and coarse nodes. Cover edges establish scale; relation edges connect source-supported directions across the graph. **2:** The question guides the agent's visits. Before and after a visit, the scene represents the same memory while moving fine detail toward the current focus and retaining coarser views elsewhere. **3:** Opening a source supplies opened source evidence for reading and review. Review can request further evidence before the answer is finalized.

4 METHOD

MemoryLOD combines a hierarchy cut, focus-dependent detail allocation, and automatic updates after movement into one scene policy (Figure 2). The resulting bounded view shows fine detail near the current focus and coarse representations of distant regions. Source openings acquire evidence under evidence character and image limits. Progress tracking and answer review direct further inspection toward unresolved question requirements.

4.1 SOURCE-GROUNDED MULTIRESOLUTION REPRESENTATION

A *source unit* is an indexed text region or a multimodal state. MemoryLOD represents the set L of source units in a multiresolution graph $G = (V, E_C \cup E_R)$. A *fine node* represents one source unit; a *group node* represents a set of source units. Cover edges E_C form the *cover hierarchy* T. Relation edges E_R connect source-supported directions such as adjacent events, shared entities, and explicit cross-references in the source content. The cover hierarchy establishes scale; relation edges provide alternative navigation paths.

Constructing the levels of detail. Source boundaries, local ordering, and cover edges determine groups. Each node has a primary parent, and large groups receive intermediate subdivisions in stable source order. Fine nodes retain source locators and a *routing digest* of informative source observations with their speaker, time, entity, or structural scope. Routing digests are constructed extractively using deterministic rules: source sentences compete on lexical diversity, structural cues, and diversity of source positions; interface records additionally retain ordered controls and observed field values. Parent routing digests select distinctive source observations from child queues in round-robin order and merge compatible contextual descriptors. This balances contributions across children while preserving the selected source observations' source bindings. Broader nodes represent more content through cues that distinguish their children (Figure 2).

Construction is independent of the question. During navigation, the question and unresolved requirements select content from each visible node's routing digest for its *routing summary*. Lexical overlap, weighted by term rarity within the node's routing digest, ranks the retained content for

display. Routing digest construction and rendering use zero language-model calls; their costs are source reading, CPU processing, and storage. Routing summaries contribute input tokens to the navigator. Routing summaries distinguish visible regions while preserving structural coverage; finer navigation and direct inspection recover detail omitted at coarser resolutions. Appendix D.2 specifies the extraction rules, aggregation, ranking, and display limits. Appendix E shows constructed graphs and their routing summaries.

Figure 3 summarizes how navigation, evidence acquisition, and review interact during question answering.

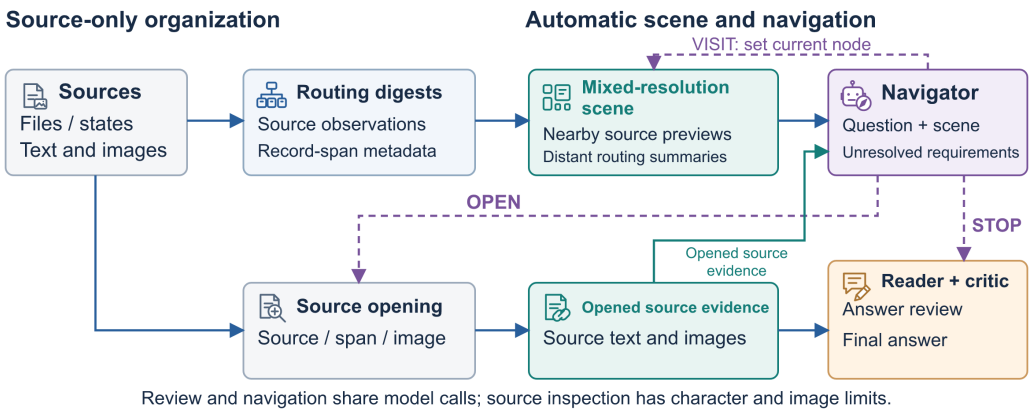


Figure 3: MemoryLOD. Source-grounded routing representations support a scene with source previews near the focus and structural coverage of distant source units. Visits change the focus; openings acquire opened source evidence. Navigation tracks inspected content and unresolved question requirements. Answer review can redirect navigation toward an evidence gap that matters to the answer, within the shared model-call budget and evidence character budget. Dashed arrows denote control.

4.2 Automatic Mixed-Resolution Scenes

The scene concentrates detail near the current focus while preserving alternative directions through a *hierarchy cut* C_t of T at navigation step t. If $M(v) \subseteq L$ denotes the source units represented by node v, then

$$\bigcup_{c \in C_t} M(c) = L. \tag{1}$$

Starting from a coarse view, refinement replaces a cut node with its child nodes. Here C is the working hierarchy cut used to construct C_t:

$$C \leftarrow (C \setminus \{c\}) \cup \mathrm{children}_T(c). \tag{2}$$

A parent's membership is the union of its children's memberships, so refinement preserves structural coverage. The scene allocates detail along the focus path and its structural neighborhood, using distance in T to define proximity and a scene card budget to limit allocation. Distant regions remain coarse. Nearby source previews and cards for relation targets supplement the hierarchy cut. Each visit changes the focus and automatically recomposes the scene. Appendix D.1 gives the structural coverage argument.

A *scene card* displays one node. Its routing summary is selected from that node's routing digest. A *source preview* is the scene card for a fine node, displaying its routing summary and identifying the source-opening target. A source preview is derived display content; `open` separately retrieves *opened source evidence* from the authoritative source. The *navigator input* includes the scene, question, opened source evidence, unresolved requirements, and remaining resource limits.

Algorithm 1 constructs the scene. Let B_{cards} be the scene card budget, K the previously revealed nodes, and R_{new} the remaining new-node allowance. The focus label is rendered separately. For a

Algorithm 1 Scene construction and update at focus f

Require: Cover hierarchy T, root r, focus f, known nodes K, previous cut C_{prev}
Limits: scene cards B_{cards}, source-preview cards B_{preview}, new nodes R_{new}
$C \leftarrow \{r\}$; $N_f \leftarrow$ bounded, ordered relation targets of f
Reserve B_{preview} source-preview card slots; $B_{\text{cut}} \leftarrow B_{\text{cards}} - |N_f| - B_{\text{preview}}$
loop
  $\mathcal{E} \leftarrow$ internal $c \in C$ on the root–focus path or with $d_T(c, f) \leq 1$,
    whose replacement $C^{(c)} = (C \setminus \{c\}) \cup \text{children}_T(c)$ satisfies $\text{Fits}(C^{(c)}, B_{\text{cut}})$
  if $\mathcal{E} = \varnothing$ **then break**
  Choose $c \in \mathcal{E}$: focus path first, then smaller $d_T(c, f)$
    Break ties by membership in C_{prev}, then stable node identifier
  $C \leftarrow C^{(c)}$
end loop
$S \leftarrow C$; append admissible targets from N_f
Append up to B_{preview} admissible source previews from the focused area
$N_{\text{new}} \leftarrow (S \cup \{f\}) \setminus K$; $K \leftarrow K \cup N_{\text{new}}$
$R_{\text{new}} \leftarrow R_{\text{new}} - |N_{\text{new}}|$; $C_{\text{prev}} \leftarrow C$
return bounded routing summaries for S and the focus label
Initially: $f \leftarrow r$, $K \leftarrow \varnothing$, $C_{\text{prev}} \leftarrow \varnothing$
After $\text{visit}(v)$**:** $f \leftarrow v$; reconstruct the scene with the retained state

proposed visible node set X_{nodes} and card limit b_{cards}, define

$$\text{Fits}(X_{\text{nodes}}, b_{\text{cards}}) \equiv |X_{\text{nodes}}| \leq b_{\text{cards}} \\ \wedge\, |(X_{\text{nodes}} \cup \{f\}) \setminus K| \leq R_{\text{new}}.$$

The scene card budget counts cards; routing-summary character limits bound their text lengths (Appendix D.2). The scene token budget bounds rendered scene tokens in the scene comparison. Model context capacity bounds the complete navigator input and output, including reasoning; it remains 262,144 tokens per request. These limits measure different resources.

The root and focus must fit the initial new-node allowance, with $B_{\text{cut}} \geq 1$. Card additions satisfy $\text{Fits}(S \cup \{v\}, B_{\text{cards}})$ and skip duplicates. Source previews sample source positions across the focused child branches; a fine-node focus supplies its own source preview. Rendering selects routing-summary content using the question and unresolved requirements. Coarse branches remain available as navigation targets alongside local detail, and opened source evidence persists as the focus moves.

4.3 Navigation and source opening

MemoryLOD separates movement from evidence acquisition through $\text{visit}(v)$, which moves to a revealed node; $\text{open}(v)$, which acquires content from a revealed fine node; and stop, which requests completion. Navigation starts at the root and uses the question, scene, opened source evidence, unresolved requirements, and remaining model-call, source-opening, evidence character, and image limits. Revealed nodes remain available for revisiting.

Opening retrieves content from the source without moving the focus. It supplies a coherent source passage containing the target, such as a record or section, with available speaker, temporal, and enclosing context. Its scope must provide enough context for interpretation while leaving room for complementary evidence. For a multimodal state, source opening returns the state text and its associated image together. Opened source evidence remains available for comparison across sources.

4.4 Evidence progress and question requirements

MemoryLOD maintains a *source inspection ledger* and *question requirements*. The source inspection ledger records which source passages or multimodal states have been opened, including complete or partial inspection of source records. Structural coverage instead records whether source units have scene representations; it does not establish source inspection. Question requirements record the needed facts, their entity and temporal scope, and their connection to evidence. Several

passages about one side of a comparison can therefore coexist with an unresolved requirement for the other side.

Requirements distinguish unresolved facts, facts with cited evidence, and premises supplied by the question. Evidence citations bind to opened excerpts; premises retain their status as assumptions. Together with prior navigation intent and critic requests, these distinctions guide inspection toward specific gaps.

4.5 Answer synthesis and bounded review

Review addresses misinterpretation, incompatible scopes, and unsupported extensions of opened source evidence. The reader synthesizes an answer from the question and opened source evidence, distinguishing source statements, premises, and deductions. A source-grounded critic examines contradictions and scope errors, and identifies evidence that could resolve a central gap.

Review can revise the answer using existing evidence or return to navigation for a missing fact. Reserved model calls, source openings, evidence characters, and image slots allow further source inspection under the original resource limits. Navigation ends on completion or exhaustion of an applicable resource limit; the answer retains supported conclusions, qualifies unresolved claims, and abstains when a central fact remains unresolved.

5 Experimental methodology

We examine answer quality and model-call use (RQ1), structural coverage and source fidelity (RQ2), variation across memory settings (RQ3), and errors remaining after source access (RQ4).

Scope and metrics. The primary evaluation covers BEAM, LongMemEval-V2, and AMA-Bench (Tavakoli et al., 2026; Wu et al., 2026; Zhao et al., 2026), with one execution per question. We use the complete 400-question BEAM 100K benchmark and the fixed, completed 10% selections for the other settings: 70 BEAM 1M questions, 46 in each LongMemEval-V2 setting, and 250 AMA-Bench questions. Limited computation resources motivate these subset evaluations. LongMemEval-V2 Small and Medium share questions; BEAM 100K and BEAM 1M use different question populations. BEAM reports fractional credit, LongMemEval-V2 native accuracy, and AMA-Bench accuracy and token F1. Inventories, model roles, and native scoring are detailed in Appendix A. We also test transfer to document question answering (Appendix C).

Comparisons and accounting. Published comparisons describe each system under its reported evaluation settings and evaluation cohorts. Matched Local Tree and Hybrid RAG comparisons test scene and retrieval policies (Appendix B). We count all navigation, reading, review, and revision calls as method costs. Source and execution replay verify provenance and consistency. Benchmark scoring remains separate from the method, with reference answers reserved for evaluation.

6 Results

Figures 4 through 6 compare MemoryLOD with published methods across the primary benchmarks. Each benchmark retains its own quality measure, and each published result retains its reported evaluation setting. Appendix A.5 distinguishes original-method references from the publications supplying the plotted scores.

Answer quality and model-call use (RQ1). On AMA-Bench, MemoryLOD achieves 64.40% accuracy and a token F1 of 0.351. Across the primary settings, MemoryLOD's mean model-call use ranges from 8.46 on AMA-Bench to 13.33 on BEAM 1M.

Scene policy and retrieval comparisons (RQ1 and RQ2). Table 1 compares final answers on matched questions. Local Tree tests the combined scene policy end to end: it replaces the mixed-resolution scene with a fixed overview and local neighborhood while retaining graph navigation and the shared answering pipeline. MemoryLOD scores higher in all four settings. Hybrid RAG

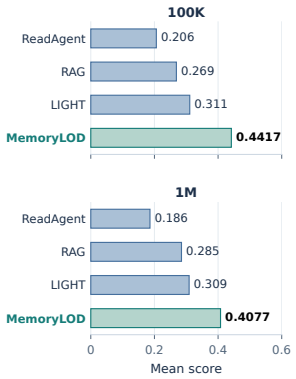


Figure 4: BEAM 100K (full) and BEAM 1M (10%) tenability mean. Published results use Qwen2.5-32B (Tavakoli et al., 2026).

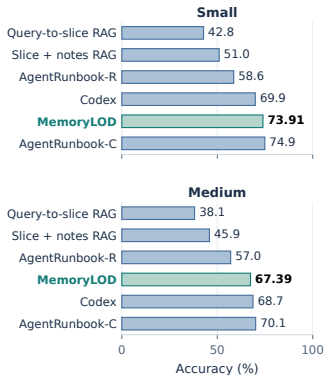


Figure 5: LongMemEval-V2 accuracy (%) on the 10% selections for Small and Medium. Published results are from Wu et al. (2026).

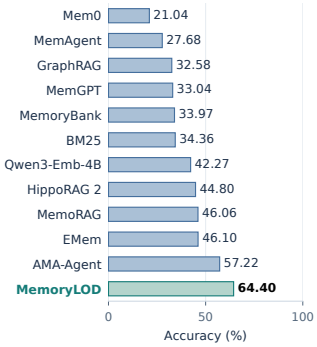


Figure 6: AMA-Bench accuracy (%) on the 10% selection. Published results use the Qwen3-32B setting of Zhao et al. (2026).

provides a system-level retrieval comparison; MemoryLOD exceeds Hybrid RAG by 36.96 percentage points on LongMemEval-V2 Small and 45.65 on Medium. The smaller gains on AMA-Bench (+1.20 points) and BEAM 100K (+0.0304 fractional credit) have paired intervals that span zero. Appendix B.1 reports the protocols and paired intervals.

Table 1: Controlled comparisons on matched questions, using the same model and reader within each setting. All scores use final answers. BEAM 100K reports fractional credit; other rows report accuracy (%).

Setting	Questions	MemoryLOD	Local Tree	Hybrid RAG
BEAM 100K	100	0.4045	0.2849	0.3741
LongMemEval-V2 Small	46	73.91	19.57	36.96
LongMemEval-V2 Medium	46	67.39	17.39	21.74
AMA-Bench	250	64.40	56.00	63.20

Evidence provenance (RQ2). Source and request replay confirm that opened source evidence reaches the reader. Scene replay confirms structural coverage, and saved-response replay reproduces native evaluation. These checks establish consistent source delivery and structural coverage throughout the recorded executions.

Memory scale and error patterns (RQ3 and RQ4). MemoryLOD achieves 73.91% accuracy on LongMemEval-V2 Small and 67.39% on Medium, a difference of 6.52 percentage points on paired questions. The two memory settings vary in both size and content; BEAM 100K and BEAM 1M also differ in question population. On BEAM 100K, instruction following (0.381) and contradiction resolution (0.175) are among the lower-scoring abilities.

7 Ablation Studies

We compare the first draft with the full post-draft stage on the primary settings. Scene composition and source opening are examined at fixed recorded navigation states and opening targets from the cross-task evaluation cohorts. Appendix B contains the complete figures, protocols, and diagnostic outcomes.

Answer review (RQ1 and RQ4). Review improves LongMemEval-V2 accuracy, yields small aggregate changes on BEAM, and reduces AMA-Bench accuracy. Appendix B.2 reports the paired results and token costs (Figure 7).

Scene composition (RQ2). Automatic and fixed-root scenes preserve full structural coverage across the scene token budget sweep. At the standard scene token budget, automatic LOD exposes more source previews near the focus; the next recorded opening target is visible in 28% of automatic scenes and 4% of fixed-root scenes. Local-only views match the former target visibility while representing 7.25% of source units (Figure 8).

Source opening (RQ2). At identical opening targets, source opening increases the complete-record count by 45.38%, at a cost of 22.66% more charged evidence characters (Figure 9).

Evidence use (RQ4). Source-linked examples identify acquisition gaps, incomplete answers, and review-induced scope changes; Appendix E illustrates the recorded retrieval behavior in interface and conversational memories.

8 Limitations and impact

Coarse routing digests can omit useful facts, and refinement around an early focus can leave relevant distant evidence behind weak routing cues. Questions spanning several branches rely on redirection within the model-call budget. Source openings provide record structure at the cost of larger, sometimes overlapping passages, which can exhaust the evidence character budget.

Answer review can change a useful interpretation: the AMA-Bench regression and the BEAM temporal example show revisions that reduce answer quality or induce abstention. Review also consumes additional model calls and input/output tokens. All review calls count toward the shared per-question model-call budget.

The scene and opening studies measure behavior at fixed recorded navigation states; the controlled comparisons test the joint effect of structural coverage and adaptive detail on answer quality. Published comparisons use different models and evaluation cohorts. Our cost measurements cover generative inference and serialized storage; construction time, retrieval computation, and end-to-end latency call for a broader efficiency study.

9 Conclusion

MemoryLOD lets an agent navigate memory at multiple resolutions through a shared interface for visits, source openings, and completion. Visits move detail toward the current focus while preserving structural coverage; source openings supply opened source evidence, and bounded answer review can return to retrieval. Answer quality and the effects of review vary by task. Studies at fixed recorded navigation states show how scene composition affects structural coverage and source-preview counts, and how source opening increases both complete-record counts and charged evidence characters.

AI use statement

Generative AI tools assisted literature research, method implementation, experiment orchestration, debugging, analysis, and manuscript editing. The authors reviewed all AI-generated content and take full responsibility for the work.

Ethics statement

The study uses existing public research benchmarks. Raw source excerpts, model-call payloads, and review packets remain access-controlled.

Reproducibility statement

The supplementary materials include the source implementations used for the reported evaluations, their hash manifests, and the subset selector with exact question identities and hash ranks. We will release the code publicly under an open-source license. The appendices describe the method's structural properties, evaluation protocol, and experimental settings.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 9112–9141, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/25f7be9694d7b32d5cc670927b8091e1-Paper-Conference.pdf.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. Walking down the memory maze: Beyond context limit through interactive reading, 2023. URL https://arxiv.org/abs/2310.05029.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory, 2025. URL https://arxiv.org/abs/2504.19413.

circlemind-ai. fast-graphrag. GitHub repository, 2024. URL https://github.com/circlemind-ai/fast-graphrag. Accessed September 25, 2026.

James H. Clark. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 19(10):547–554, October 1976. ISSN 1557-7317. doi: 10.1145/360349.360354. URL http://dx.doi.org/10.1145/360349.360354.

Jihao Dai, Dingjun Wu, Yuxuan Chen, Zheni Zeng, Yukun Yan, Zhenghao Liu, and Maosong Sun. Navirag: Towards active knowledge navigation for retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2604.12766.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025. URL https://arxiv.org/abs/2404.16130.

David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, and James W. O'Toole. Semantic file systems. In *Proceedings of the thirteenth ACM symposium on Operating systems principles*, SOSP91, pp. 16–25. ACM, September 1991. doi: 10.1145/121132.121138. URL http://dx.doi.org/10.1145/121132.121138.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569. Curran Associates, Inc., 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Hugues Hoppe. Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH96, pp. 99–108. ACM, August 1996. doi: 10.1145/237170.237216. URL http://dx.doi.org/10.1145/237170.237216.

Haoyu Huang, Yongfeng Huang, Yang Junjie, Zhenyu Pan, Yongqiang Chen, Kaili Ma, Hongzhi Chen, and James Cheng. Retrieval-augmented generation with hierarchical knowledge. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6044–6060, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.321. URL https://aclanthology.org/2025.findings-emnlp.321/.

Edward Izgorodin. Semantic level of detail for knowledge graphs: Discovering abstraction boundaries via spectral heat diffusion, 2026. URL https://arxiv.org/abs/2603.08965.

Shuo Ji, Yibo Li, and Bryan Hooi. Memory is reconstructed, not retrieved: Graph memory for llm agents, 2026. URL https://arxiv.org/abs/2606.06036.

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 Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26396–26415. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/lee24c.html.

Letta. Announcing Letta, September 2024. URL https://www.letta.com/blog/announcing-letta/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, Wenbo Su, and Bo Zheng. GraphReader: Building graph-based agent to enhance long-context abilities of large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12758–12786, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.746. URL https://aclanthology.org/2024.findings-emnlp.746/.

Lei Liang, Mengshu Sun, Zhengke Gui, Zhongshu Zhu, Zhouyu Jiang, Ling Zhong, Yuan Qu, Peilong Zhao, Zhongpu Bo, Jin Yang, Huaidong Xiong, Lin Yuan, Jun Xu, Zaoyang Wang, Zhiqiang Zhang, Wen Zhang, Huajun Chen, Wenguang Chen, and Jun Zhou. Kag: Boosting llms in professional domains via knowledge augmented generation, 2024. URL https://arxiv.org/abs/2409.13731.

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

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. 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 on Web Conference 2025*, WWW '25, pp. 2366–2377. ACM, April 2025. doi: 10.1145/3696410.3714805. URL http://dx.doi.org/10.1145/3696410.3714805.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 990–1023, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/0382cb76309820f71c6eacd47b36ce71-Paper-Conference.pdf.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32628–32649, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8a2acd174940dbca361a6398a4f9df91-Paper-Conference.pdf.

Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J Mitchell. Beyond a million tokens: Benchmarking and enhancing long-term memory in llms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 133343–133429, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/d7f0cfa0fe759b033d5262e1bb7d4065-Paper-Conference.pdf.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

Di Wu, Zixiang Ji, Asmi Kawatkar, Bryan Kwan, Jia-Chen Gu, Nanyun Peng, and Kai-Wei Chang. Longmemeval-v2: Evaluating long-term agent memory toward experienced colleagues, 2026. URL https://arxiv.org/abs/2605.12493.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 17577–17604. Curran Associates, Inc., 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/19909c36f51abc4856b4560aff3d36d6-Paper-Conference.pdf.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent, 2026. URL https://arxiv.org/abs/2507.02259.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025. URL https://arxiv.org/abs/2506.05176.

Yaoze Zhang, Rong Wu, Pinlong Cai, Xiaoman Wang, Guohang Yan, Song Mao, Ding Wang, and Botian Shi. Leanrag: Knowledge-graph-based generation with semantic aggregation and hierarchical retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(41): 34862–34869, March 2026. ISSN 2159-5399. doi: 10.1609/aaai.v40i41.40789. URL http://dx.doi.org/10.1609/aaai.v40i41.40789.

Yujie Zhao, Boqin Yuan, Junbo Huang, Haocheng Yuan, Zhongming Yu, Haozhou Xu, Lanxiang Hu, Abhilash Shankarampeta, Zimeng Huang, Wentao Ni, Yuandong Tian, and Jishen Zhao. Ama-bench: Evaluating long-horizon memory for agentic applications, 2026. URL https://arxiv.org/abs/2602.22769.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731, March 2024. ISSN 2159-5399. doi: 10.1609/aaai.v38i17.29946. URL http://dx.doi.org/10.1609/aaai.v38i17.29946.

Sizhe Zhou, Sheldon Yu, Hui Wei, Junda Wu, Siru Ouyang, Yizhu Jiao, Shijia Pan, Julian McAuley, Yu Zhang, Tong Yu, and Jiawei Han. Filesystem-based memory for llm agents: Organization, evolution, and sustainability, 2026. URL https://arxiv.org/abs/2607.26637.

A Evaluation protocol and reproducibility

A.1 Inventories and model roles

The reported results use the full 400-question BEAM 100K benchmark and the completed, fixed 10% selections for all other settings: 70 of 700 BEAM 1M questions, 46 of 451 questions in each LongMemEval-V2 setting, and 250 of 2,496 AMA-Bench questions. Integer rounding makes the LongMemEval-V2 and AMA-Bench selections slightly larger than 10%. Across these five cohorts, we run MemoryLOD 812 times, producing one final answer per selected question. LongMemEval-V2 Small and Medium share question identities; BEAM 100K and BEAM 1M have separate question populations. The source-only index remains fixed during each execution. Reference answers and judge outputs are reserved for evaluation.

MemoryLOD's navigator, reader, and critic use Qwen3.8-27B (Qwen Team, 2026). Published comparator systems retain their respective answer models, and benchmark judges serve a separate assessment role. Matched comparisons bind question identities and scoring profiles.

Subset selection. Limited computation resources motivate the 10% evaluations. For a setting with N questions, we select $n_{\text{questions}} = \lceil 0.1N \rceil$ using proportional stratification: BEAM and AMA-Bench by question type, LongMemEval-V2 by (domain, question type). For stratum h with N_h questions, assign $\lfloor n_{\text{questions}} N_h / N \rfloor$ places, then allocate the remaining places by decreasing fractional remainder, breaking ties by the lexicographic stratum key. Within each stratum, rank question identifiers by SHA-256 of a fixed seed concatenated with the identifier and take the assigned number. Small and Medium reuse the same 46 LongMemEval-V2 identifiers; BEAM 1M contains seven questions per ability. The selector uses identifiers and strata independently of answer quality. We compare methods on the same fixed question identities. The supplement provides the hash ranks, selector, and question identities. BEAM 100K uses all 400 questions; the balanced 100-question controlled comparison is a separate cohort (Appendix B.1).

A.2 Experimental conditions

All MemoryLOD model roles use reasoning and a model context capacity of 262,144 tokens per request. Sampling uses temperature 0.6, top-p 0.95, and top-k 20. Navigation and critic calls allow up to 32,768 output tokens. Reader calls, including revision, allow 32,768 output tokens for text benchmarks and 20,000 for LongMemEval-V2. Table 2 specifies the per-question model-call budget, source-opening limit, evidence character budget, and evidence image limit. Further source openings and answer revisions consume the remaining amounts of these same resources.

Table 2: Per-question resource limits. Navigation, reading, and review share the model-call budget. Evidence character and image limits bound opened source evidence; each request retains the full model context capacity.

Resource	Text	Multimodal
Model-call budget (calls)	25	25
Source-opening limit (openings)	8	8
Evidence character budget (characters)	80,000	200,000
Evidence image limit (images)	—	8
Critic-call limit (calls)	1	1
Reader-call limit, including revision (calls)	2	2

A.3 Native scoring

BEAM uses GPT-4.1-mini at temperature zero with the canonical question and rubric, fractional credit, and native normalized Kendall correlation for event ordering. The ten ability means receive equal weight. Each ability contains 40 evaluated questions at 100K and seven in the 1M selection. LongMemEval-V2 retains its native deterministic evaluators and GPT-5.2 judgments where prescribed, with medium reasoning and a 4,096-completion-token limit for the judge.

AMA-Bench uses its prescribed Qwen3-32B external judge at temperature zero with a 2,048-output-token limit. Native binary judgments determine accuracy, and the native token-overlap evaluator provides mean per-question F1. Every grade remains bound to its question, generated answer, and benchmark-specific reference answer.

Model-based judgments introduce variability into score differences, particularly when the observed changes are small (Zheng et al., 2023).

A.4 Audit and cost scope

Each completed answer undergoes source and execution checks before native scoring. Replay reconstructs source openings, displayed scenes, and reader inputs from fixed inputs and saved responses; grading replay verifies the association between judge responses and reported scores. Scoring proceeds independently for completed answers, with reference answers and scoring rubrics isolated from method construction and navigation.

Method-call costs include navigation, reading, criticism, and revision. Call means use the reported evaluation cohort for each setting, and all recorded attempts contribute to cost. Input and output token accounting includes reasoning output. Construction, source audits, scene rendering, storage, elapsed time, and external judging retain separate cost categories. Quality and cost comparisons preserve the corresponding question population and scorer definition.

A.5 Additional baseline references and score provenance

Table 3 credits original methods that appear in the comparison figures and tables. The score-source column identifies the publication whose evaluation supplies the plotted value; the original-method citation identifies the underlying approach. The remaining named methods are cited in Related Work or in the corresponding benchmark paper.

Table 3: Original references and sources of published comparison scores.

Method	Original reference	Score source
Mem0	Chhikara et al. (2025)	Zhao et al. (2026)
MemAgent	Yu et al. (2026)	Zhao et al. (2026)
MemoryBank	Zhong et al. (2024)	Zhao et al. (2026)
MemoRAG	Qian et al. (2025)	Zhao et al. (2026)
Qwen3-Emb-4B	Zhang et al. (2025)	Zhao et al. (2026)

The EMem score is reproduced from Table 5 of Zhao et al. (2026). That paper attributes EMem to an anonymous reference with a bibliographic placeholder. The original publication has not been identified, so we attribute the plotted result to AMA-Bench.

B Controlled comparisons and ablations

B.1 Controlled comparisons

Selection and shared conditions. We compare the methods on 442 questions: 100 from BEAM 100K, 46 from each LongMemEval-V2 setting, and 250 from AMA-Bench. The BEAM 100K cohort has ten questions per ability and five from each of 20 conversations; the other settings use the fixed 10% selections. We reuse MemoryLOD's saved final answers and generate one answer per question with Local Tree and Hybrid RAG. All 884 control answers pass source, execution, and native scoring replay, with no method or judge failures. All arms share the model and generation settings in Appendix A.2, source units, source opening, reader instructions, and evidence character and image limits in Table 2. MemoryLOD and Local Tree also share the review policy and model-call budget.

Local Tree. Local Tree replaces automatic mixed-resolution scenes with a fixed overview and the local tree neighborhood around the current focus. All source units remain reachable through the

same navigation actions. Comparing final answers tests the scene policy while retaining the rest of the method. Both methods use routing summaries to guide access and opened source evidence to support answers. On BEAM 100K, MemoryLOD scores 0.1196 higher on average (95% interval [0.0603, 0.1872]).

Hybrid RAG. Hybrid RAG combines BM25 with dense retrieval using `BAAI/bge-small-en-v1.5`, retrieving 32 candidates per channel and combining them through reciprocal-rank fusion ($k_{\mathrm{RRF}} = 60$). The `jinaai/jina-reranker-v1-turbo-en` cross-encoder reranks the candidates before source opening. A single call to the shared reader produces the answer. We compare this answer with MemoryLOD's final answer to assess end-to-end answer quality. On BEAM 100K, MemoryLOD's mean score is 0.0304 higher (95% interval $[-0.0394, 0.1023]$). This comparison includes MemoryLOD's navigation and review stages.

For LongMemEval-V2, Hybrid RAG searches the source text of recorded states and passes the selected states and screenshots to the reader, using the same evidence character and image limits.

Scoring and uncertainty. BEAM scores average fractional credit equally across the ten abilities; LongMemEval-V2 and AMA-Bench use native accuracy. Table 4 gives BEAM 100K's per-ability scores. We draw 5,000 bootstrap samples of source collections with replacement and compute the question-weighted mean difference for each sample. We report pointwise percentile 95% intervals. Each of the 20 BEAM 100K conversations contributes five questions. For LongMemEval-V2 Small, resampling is limited to its two source collections. Table 5 reports the paired differences for every setting.

Table 4: BEAM 100K per-ability fractional credit on 100 questions (ten per ability). All methods are evaluated on their final answers.

Ability	MemoryLOD	Local Tree	Hybrid RAG
Abstention	0.9000	0.9000	0.7000
Contradiction resolution	0.1500	0.1750	0.0500
Event ordering	0.1907	0.2119	0.1450
Information extraction	0.5167	0.2917	0.6083
Instruction following	0.5000	0.3750	0.4500
Knowledge update	0.3000	0.3000	0.4000
Multi-session reasoning	0.3250	0.1000	0.3375
Preference following	0.6250	0.3250	0.5500
Summarization	0.2375	0.0200	0.2750
Temporal reasoning	0.3000	0.1500	0.2250

Table 1 reports the BEAM 100K, LongMemEval-V2, and AMA-Bench scores. MemoryLOD scores higher than Local Tree in all four settings.

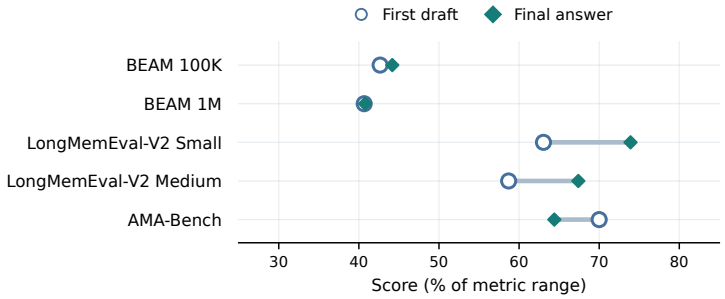
Table 5: MemoryLOD final minus control, with paired 95% source-cluster bootstrap intervals. Accuracy differences are percentage points; BEAM retains fractional-credit units.

Setting	Collections	Δ Local Tree [95% interval]	Δ Hybrid RAG [95% interval]
BEAM 100K	20	$+0.1196$ [$+0.0603$, $+0.1872$]	$+0.0304$ [-0.0394, $+0.1023$]
LME Small	2	$+54.35$ [$+54.17$, $+54.55$]	$+36.96$ [$+31.82$, $+41.67$]
LME Medium	46	$+50.00$ [$+32.61$, $+65.22$]	$+45.65$ [$+28.26$, $+60.87$]
AMA-Bench	152	$+8.40$ [$+1.59$, $+15.02$]	$+1.20$ [-5.51, $+8.47$]

B.2 Contribution of answer review (RQ1 and RQ4)

We compare the first draft with the final answer to measure the combined effect of criticism, corrective evidence acquisition, and revision after a shared navigation prefix.

Review substantially improves LongMemEval-V2 scores, changes BEAM aggregates little, and reduces AMA-Bench scores (Figure 7). The post-draft stage consumes 25.03% of reported input and output tokens over these 812 paired executions.



	First draft	Final answer
BEAM 100K	0.4265	0.4417
BEAM 1M	0.4066	0.4077
LongMemEval-V2 Small	63.04%	73.91%
LongMemEval-V2 Medium	58.70%	67.39%
AMA-Bench	70.00%	64.40%

Figure 7: Answer quality at the first draft and after review. Cohorts are the full BEAM 100K benchmark and the completed 10% selections for the other settings. Positions express scores as a percentage of the metric range: BEAM scores are multiplied by 100, and accuracy is expressed in percent. The aligned columns retain native scores. Comparisons follow the benchmark-specific metric within each row.

The first draft is the first usable reader answer. The answer-stage comparison retains each trajectory through its first draft and evaluates that draft with the same native scoring profile used for the final answer. Identical answers retain their original scores. The cost difference comprises all subsequent review, navigation, and answer revision. The contrast measures the combined post-draft stage.

B.3 Contribution of mixed-resolution scenes (RQ2)

To characterize structural coverage and source-preview allocation, we compare automatic scenes with a *fixed-root* view anchored at the graph root and a *local-only* view containing source previews around the focus. For each recorded navigation state, the variants share a scene token budget and a common focus description. We vary the scene token budget over $0.5\times$, $1\times$, and $2\times$ the recorded scene's token count. The comparison measures represented source units, source-preview counts, and actual scene token counts at the same navigation states.

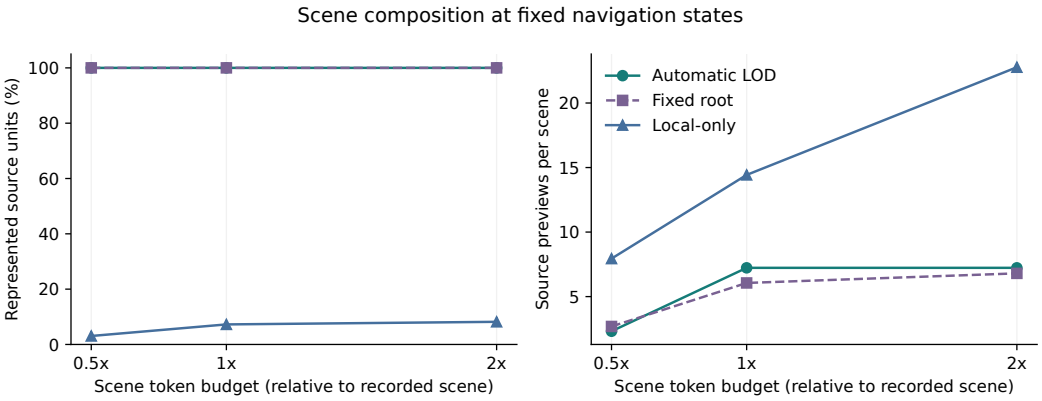


Figure 8: Structural coverage and source-preview counts under a common scene token budget. Automatic and fixed-root views maintain full representation of source units. Local-only views devote more of the scene to fine detail within a smaller represented region. The two curves for full structural coverage overlap.

Automatic and fixed-root scenes preserve 100% structural coverage throughout the scene token budget sweep (Figure 8). At the standard scene token budget, local-only views represent 7.25% of source units, with an average of 14.43 source previews per scene. Automatic LOD exposes 7.23 source previews, compared with 6.06 for fixed-root views. Within the current focus area, the corresponding averages are 7.23 and 2.60, respectively.

We also examine whether the next recorded opening target appears at fine resolution. At the standard scene token budget, this target is visible in 28% of eligible automatic and local-only scenes, compared with 4% of fixed-root scenes. Automatic scenes therefore retain structural coverage while exposing more locally useful detail than fixed-root scenes along the recorded trajectories.

Scene analysis uses question-identity-based selection within each benchmark setting and samples the beginning, middle, and end of the available scene sequence. Scene variants share the focus, cover hierarchy, scene token budget, and presentation conditions. Actual scene token counts vary with the cards each scene admits: at the standard scene token budget, automatic, fixed-root, and local-only views consume an average of 6,679, 5,120, and 4,750 tokens, respectively. At half the scene token budget, automatic LOD exposes 2.31 source previews per scene, compared with 2.69 for fixed-root views. At twice the scene token budget, the corresponding averages are 7.23 and 6.80. Automatic detail saturates because of the proximity and source-preview rules.

B.4 Contribution of source opening (RQ2)

We compare the original indexed region with the coherent source passage opened by the method, holding the target sequence fixed. The *complete-record count* counts source records whose full text appears in the inspected passages. The *partial-record count* counts source records with some, but not all, of their text inspected. Each record is counted once per question instance, and counts are then summed across question instances. These are counts, not structural coverage percentages. Multimodal access retains its atomic source states.

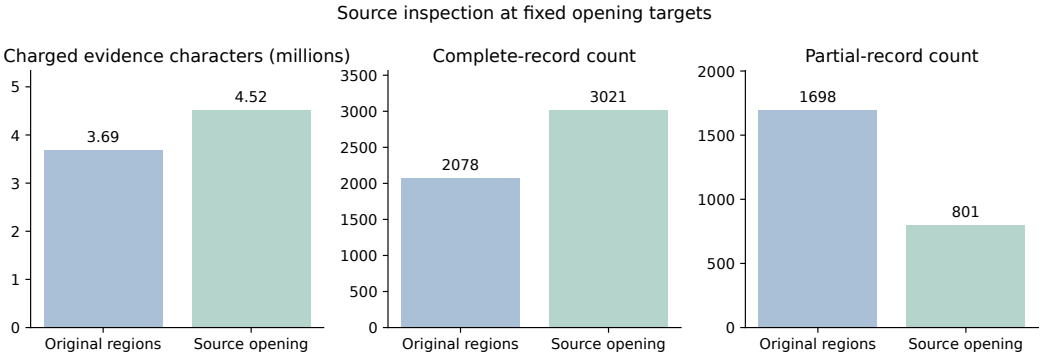


Figure 9: Charged evidence characters and record counts for the same opening targets. Source openings increase the complete-record count and reduce the partial-record count while consuming more evidence characters. Aggregates sum across question instances.

Source openings increase the complete-record count by 45.38%, while increasing charged evidence characters by 22.66% (Figure 9). The partial-record count decreases from 1,698 to 801.

The *unique inspected-character count* counts the union of opened source spans within each question instance and sums their lengths across question instances. It increases from 3,492,189 to 4,122,121 characters. The *charged evidence-character count* instead counts all supplied text, including headings and repeated passages, against the evidence character budget. Its larger increase reflects both added passage context and overlap between source openings.

C Transfer to document QA

We evaluate UltraDomain on 43 of 430 questions (10%): ten each in agriculture, computer science, and law, and thirteen in the mixed domain. We stratify selection by domain and use the hash-ranked selection procedure and text-benchmark conditions in Appendix A. The native LeanRAG Overall evaluator uses DeepSeek-V3-0324 with five accepted votes per answer; the aggregate weights the 43 questions equally. Table 6 compares MemoryLOD with published methods under their reported settings.

Table 6: UltraDomain Overall quality on a ten-point scale. MemoryLOD is evaluated on 43 questions; published comparison scores retain the evaluation settings of Zhang et al. (2026). Published aggregates weight rounded domain scores by question frequency.

Method	Overall (/10)
FastGraphRAG (circlemind-ai, 2024)	4.74
KAG (Liang et al., 2024)	7.72
LightRAG (Guo et al., 2025)	8.09
NaiveRAG	8.23
GraphRAG (local) (Edge et al., 2025)	8.35
HiRAG (Huang et al., 2025)	8.41
LeanRAG (Zhang et al., 2026)	8.69
MemoryLOD	8.00

Matched controls. On the same 43 questions, MemoryLOD's final answers score 8.005, Local Tree 5.005, and Hybrid RAG 8.293. MemoryLOD's paired differences are $+3.000$ against Local Tree (95% interval [1.805, 4.251]) and -0.288 against Hybrid RAG ([-0.470, -0.114]). The intervals come from 5,000 question-weighted bootstrap replicates over four source collections. Both controls complete all 43 executions, pass native scoring replay, and have no method or judge failures.

D Method details and structural properties

D.1 Structural properties and scope

Structural coverage preservation. Assume that the rooted hierarchy covers L and that every internal node's membership is the union of its children's memberships. The initial cut consisting of the root covers L. Replacing a cut node by all its children preserves this union. By induction, every refinement permitted by the scene card budget and new-node allowance preserves Equation 1. Regions awaiting refinement remain represented at their coarser resolution. Additional source previews and relation-target cards preserve structural coverage.

Source fidelity. Opening supplies content from the underlying source, preserving its record context and the association between text and images in a multimodal state. Source locators identify the content inspected by the agent.

Resource limits. Navigation, synthesis, and review consume a shared model-call budget. Returning from review to navigation does not reset that budget. Source-opening limits, evidence character budgets, and evidence image limits separately bound source inspection. Model context capacity remains fixed for every request.

Semantic scope. Structural coverage guarantees representation of source units. Answering also requires the relevant facts to survive compression and the reader to interpret opened passages at the question's scope. Claims of global absence require evidence covering that scope.

D.2 Routing digest construction and rendering

A *routing digest* stores source observations and structural descriptors; a *routing summary* selects routing-digest content for the current question. A fine node's scene card presents its routing summary as a *source preview*. Only source opening supplies *opened source evidence*. Construction is deterministic and query-independent. Only display ranking uses the question and missing-evidence request. Routing summaries and source previews guide navigation; answers use opened source evidence.

D.2.1 Source features and context

Inputs are indexed text regions or multimodal states, their stable source identifiers and order, index metadata, and a cover hierarchy with ordered children. Aggregation follows this hierarchy; other

graph relations support navigation. Text extraction reads the indexed source substring. Multimodal extraction reads the URL, accessibility text, and recorded action; images remain attached to the state for source opening.

Source observations retain exact source text, fine-node identifiers, and half-open Unicode-character offsets. Descriptors retain a kind, scope, ordered values, support spans, and occurrence count; their identity is $(\text{kind}, \text{scope}, \text{values})$. Sorting is stable, with source offsets or identifiers resolving ties. Let $W(x)$ contain case-folded lexical tokens (words longer than two characters), and let $U(x)$ remove the fixed English stopword list. Extraction and aggregation use W; display ranking uses U.

Source observation selection. Split source text into sentence candidates of 4–420 characters and retain the first occurrence of each case-folded, whitespace-collapsed sentence. Let $f_{\text{sent}}(w)$ count candidates containing term w. Divide the source into four position bands. If more than 96 candidates remain, keep up to 24 per band by the score below with $W_{\text{seen}} = \varnothing$ and without the band bonus, then restore source order. Retain the original frequencies for subsequent selection. Starting with previously selected terms $W_{\text{seen}} = \varnothing$ and represented bands $J = \varnothing$, greedily maximize

$$S_{\text{extract}}(x_i) = \frac{\sum_{w \in W(x_i) \setminus W_{\text{seen}}} f_{\text{sent}}(w)^{-1/2}}{\sqrt{\max(6, |W(x_i)|)}} + h_i + z_i + \mathbf{1}[b_i \notin J]. \tag{3}$$

Here b_i is the candidate's position band; $h_i = 2$ for a recognized heading, numbered item, or structural header and zero otherwise, while $z_i = 0.7$ for candidates containing a digit. Update W_{seen} and J after each selection, prefer earlier source offsets on ties, and return source observations in source order. Select 12 source observations when a page descriptor is present. Otherwise select up to 36 informative source observations, requiring at least three non-stopword terms and excluding heading and metadata-only prefixes; fall back to the initial 12 if none qualify.

Structure and attribution. Source terms are the first whole-word source matches of indexed dates, entities, and terms, capped at 24 distinct values with their original spellings and spans. Structural extraction preserves Markdown headings and table-column order, accessibility page and heading labels, row labels, tabs, links, buttons, editable controls, and observed field values. Labels are deduplicated within their kind and scope while retaining source order. Named interface containers contribute scope through the page and last two enclosing containers. Field values retain their field/child support; recorded transitions link an action to its target label in the preceding state.

For text, CommonMark heading ancestry supplies context. The largest repeated family of numbered headings defines record boundaries, with ties resolved by shallower level and first encounter. Recurring direct-child headings identify record fields. Each source observation receives its heading path, record ordinal, and identity $(\text{source ID}, \text{record start}, \text{exact text})$. Complete heading bodies of at most 420 characters supplement the selected source observations. Descriptors are extracted within record-scoped intervals and carry the same heading context. *Record-span metadata* unions overlapping or touching intervals by source identifier and record start. This construction metadata does not record agent inspection. For non-page source observations, recognized user/assistant headings provide speaker attribution; text-record context replaces this initial speaker-only wrapper.

Fine-node titles may include the source page label. Non-root parent titles use up to three distinct child page labels (section labels if no pages are available), sorted lexicographically; other supplied titles remain intact. Parser expressions, label-cleaning rules, and exact tie-breaking keys are specified in the supplementary implementation, including `lod_digest.py` and `lod_source_context.py`.

D.2.2 Bottom-up aggregation

Process nodes in postorder. Let $f_{\text{child}}(w)$ count children whose source observations contain term w. Within each child, rank source observations by decreasing

$$S_{\text{aggregate}}(x) = \frac{\sum_{w \in W(x)} 1/\max(1, f_{\text{child}}(w))}{\sqrt{\max(1, |W(x)|)}}, \tag{4}$$

then by source-node identifier and offset. Traverse child queues round-robin, taking the first unseen source observation from each until 64 source observations are retained or the queues are exhausted.

Deduplication uses text-record identity when available and normalized text otherwise; source support and context remain attached.

Split independently meaningful labels (buttons, links, editable controls, headings, row labels, and source terms) into single-value descriptors; ordered structural tuples remain intact. Merge equal descriptor identities, sum occurrence counts, and retain distinct support spans from the two lexicographically smallest source-node identifiers. Return descriptors in identity order and union record-span metadata across children. Only the source observation pool has a fixed cap; descriptor and record-span inventories depend on source structure.

D.2.3 QUERY-CONDITIONED DISPLAY

Concatenate the question and current missing-evidence request to form the *routing query* q_{route}. For a node with m retained source observations, let $f_{\mathrm{digest}}(w)$ count source observations containing w after stopword removal. Rank content by

$$S_{\mathrm{display}}(X_{\mathrm{words}}, q_{\mathrm{route}}) = \sum_{w \in X_{\mathrm{words}} \cap U(q_{\mathrm{route}})} \log\left(2 + \frac{m}{1 + f_{\mathrm{digest}}(w)}\right). \tag{5}$$

A source observation uses $X_{\mathrm{words}} = U(x)$. A descriptor uses its highest-scoring scope–value pair; a source term uses its own text. Absent terms have $f_{\mathrm{digest}}(w) = 0$. Let s be a proposed display line and $\mathcal{L}_{\mathrm{lines}}$ the accepted display lines. The routing-summary character limit is B_{chars}. Proposed display lines must be nonempty, distinct, and satisfy

$$|s| + \sum_{\ell \in \mathcal{L}_{\mathrm{lines}}} (|\ell| + 1) \le B_{\mathrm{chars}}, \qquad B_{\mathrm{chars}} = 1{,}100. \tag{6}$$

Skip lines that do not fit and continue. Display text is whitespace-collapsed; retained source text and offsets are unchanged.

Text records. Display up to eight relevant source terms, with their joined values capped at $\min(210, \lfloor B_{\mathrm{chars}}/4 \rfloor)$ characters. Consider one structural descriptor, or two when page information exists, requiring positive relevance and at most 240 characters per descriptor. Field values, field text, and scoped controls receive tie preference. Rank source observations by relevance, explicit query matches to record/turn/event/step ordinals, user attribution for personal questions, and source identifier/offset. Prefix them with heading context (speaker if context is empty), stopping after an attempted addition once seven lines have accumulated. A record-count footer uses the same routing-summary character limit. Empty output falls back to the first B_{chars} title characters.

Multimodal states. Initialize with up to eight source terms. Consider three descriptors with page information and one otherwise, allowing at most 400 characters each. After relevance, prioritize observed field values and scoped links, then scoped headings/controls, recorded transitions, table columns, tabs, page/editable controls, headings, sections, and remaining kinds. Rank source observations by relevance, user attribution on personal questions, informativeness, term density, and source identifier/offset. Without page information, only informative source observations are eligible. Stop after an attempted addition once six lines have accumulated. Topic initialization precedes character-limited additions, and empty output uses the full node title; these two cases retain their stated lengths rather than the routing-summary character limit.

Long descriptors. Render scope, kind, and values in their stored order. If they exceed the descriptor character limit, anchor a window at the earliest value with maximal query-word overlap, then add neighboring values by increasing distance, left first, while reserving space for a partial-display notice and stored-label count. The outer line-admission rule still applies. Exact formatting and parser constants are provided in the evaluated implementation.

Structural coverage remains stable as the scene token budget varies from $0.5\times$ to $2\times$ its standard value (Figure 8). Future work will examine how individual source observation limits, descriptor counts, and routing-summary character limits affect answer quality.

D.3 Construction and inference cost scope

Routing digest extraction, aggregation, and display ranking are CPU operations with zero additional language-model calls and zero generation tokens. Their inputs include the source index and its metadata, whose preparation has a separate cost. Reading a routing summary contributes input tokens to the Qwen3.8-27B navigator. Source loading, parsing, descriptor storage, and aggregation contribute construction costs; query ranking contributes per-rendering CPU work. The evaluation's method-call and token totals measure language-model use. LOD construction is query-independent CPU preprocessing, whose cost can be amortized across questions by caching the initialized graph for an unchanged source collection and construction configuration. Construction cost depends on collection size and content.

E Retrieval examples and error analysis

Figures 10 and 11 show actual graph neighborhoods, recorded navigation, and source-bound routing-summary excerpts from interface and conversational memories. The traces show how agents use multiresolution routing to narrow a search and inspect evidence in context.

LongMemEval-V2: locating a structured workflow. The question asks for the stages preceding the *Summary* interface stage in Magento's *Create Product Configurations* workflow. The agent follows progressively narrower areas and opens a recorded configuration state for the Nona Fitness Tank. Ordered stage labels remain visible throughout the route, allowing the agent to recognize the relevant workflow even in a coarse routing summary. Opening the state then grounds the correct answer: *Select Attributes*, *Attribute Values*, and *Bulk Images & Price*. Preserving interface scope and order across resolutions lets the agent follow a coarse cue to the precise source state. Adjacency links also connect consecutive states across cover boundaries.

BEAM 1M: gathering complementary temporal evidence. The question asks for the interval between setting a Sprint 1 deadline and adjusting it to mid-February. The agent first follows a deadline-adjustment source observation into a local region, then redirects to a broader scheduling area and opens a passage describing the earlier sprint plan. Coarse routing summaries expose relevant temporal cues across the large history, while source previews preserve the session, turn, and speaker needed to distinguish a user's adjustment from an assistant's schedule. The agent can search separate parts of memory and inspect the retrieved passages in context. During review, however, it distinguishes deadline values from the dates of the planning actions and abstains. Finding the passages leaves this question of interpretation unresolved.

Implications for retrieval. In these examples, coarse cues guide region selection and refinement exposes specific records. Source opening preserves the context needed to interpret them. The BEAM agent also redirects its search after exploring a local region.

E.1 Outcome-stratified diagnostic cases

The qualitative panel selects examples by question identity within the lower and upper score-ranked halves of each setting. Ties can place correct answers in either half. The cases trace errors in counting, interpretation, and answer completeness alongside successful retrieval:

- A BEAM multi-session answer aggregates many source concerns while the reference answer uses a narrower count. This discrepancy leaves ambiguity about counting granularity and reference-answer scope. An AMA query-performance example combines evidence of successful and unsuccessful limited queries, so its broad necessity claim is in tension with the scope supported by the opened records.
- The BEAM temporal trace above ends in abstention after the draft gives a five-day interval and review changes its interpretation. In another temporal question, the answer gives the correct interval but omits date anchors required by the rubric, receiving partial credit. A separate BEAM example correctly abstains about participants' feelings given the available records.

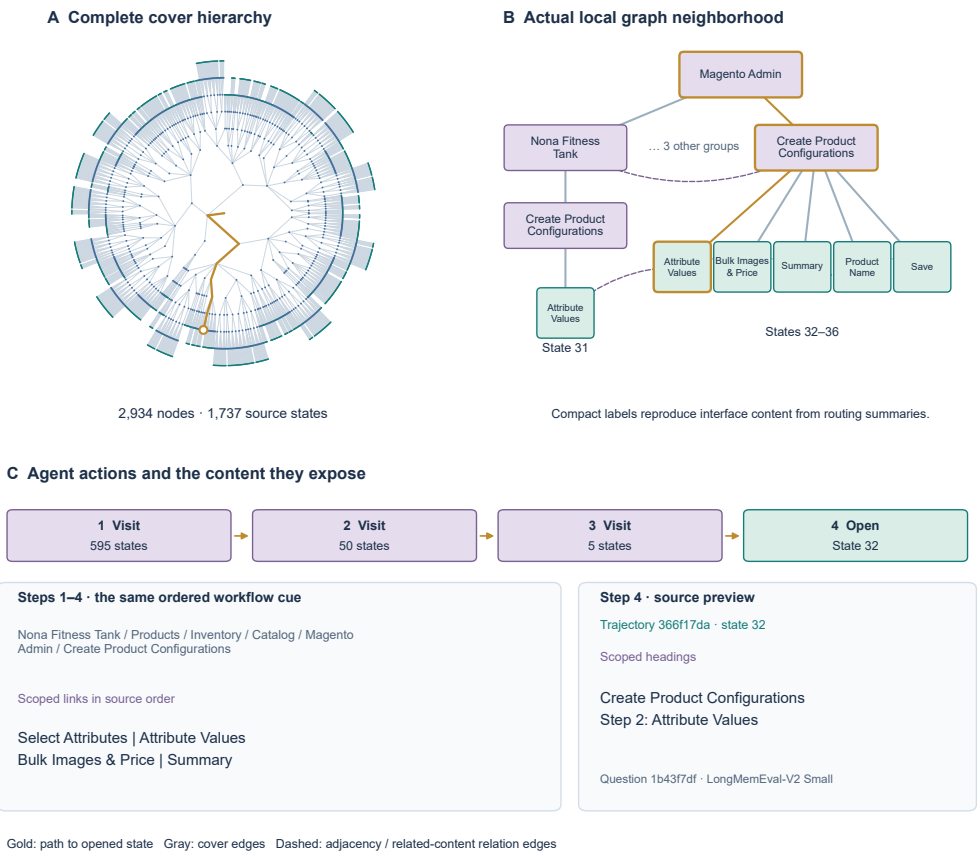


Figure 10: A real LOD graph and its routing content from LongMemEval-V2 Small, question `1b43f7df`. **A:** The complete cover hierarchy, with the path to the opened source state highlighted. **B:** An enlarged neighborhood showing actual cover edges and adjacency/related links across and within branches. Compact labels reproduce interface content from routing summaries. Lavender nodes represent groups and teal nodes represent source states; gold outlines mark the selected path. Surrounding branches are abbreviated. **C:** The recorded visit/open sequence and verbatim routing-summary excerpts, with line wrapping for readability. The common excerpt occurs at each selected resolution. The final heading excerpt describes trajectory `366f17da`, state 32.

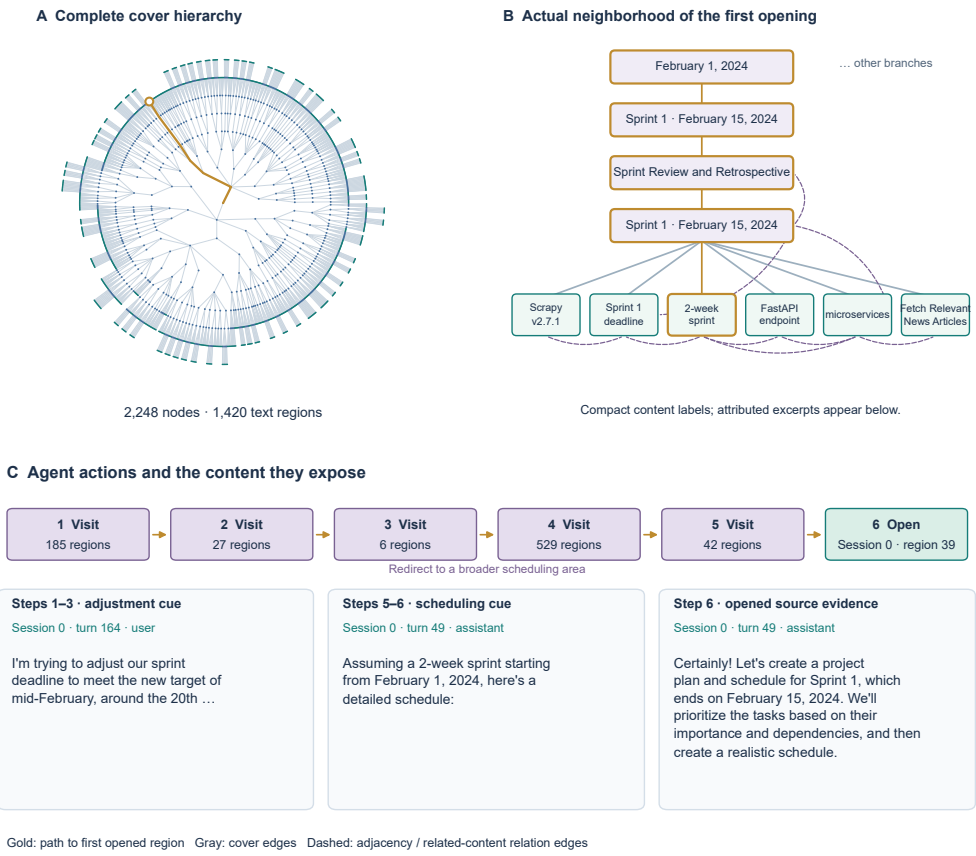


Figure 11: A real LOD graph from BEAM 1M, Conversation 2, question `2-temporal_reasoning-0`. **A:** The complete cover hierarchy, with the path to the first opened text region highlighted. **B:** Its actual local neighborhood, with compact content labels drawn from the routing summaries. Lavender nodes represent groups; teal nodes represent source regions, with the opened region outlined in gold. Additional branches are abbreviated, and dashed edges show recorded adjacency and related-content relations. **C:** The recorded route through the first opening and exact excerpts from routing summaries, source previews, and opened source evidence. The coarse routing-summary excerpt ends at the ellipsis; the middle excerpt occurs in both the selected intermediate routing summary and the source preview. The radial and local layouts arrange the constructed topology for readability.

- Successful LongMemEval-V2 examples recover an integration field, an ordered workflow, and a comparison of page capacities. A reports-procedure answer benefits from removing an unsupported claim about the destination of a click. An AMA example preserves the prerequisites for a conditional synthesis.
- A LongMemEval-V2 comparison supports the observed capacities while inferring their default status from the recorded state.

The panel provides case-level diagnoses.

F STORAGE OVERHEAD

We measure serialized LOD caches for 253 distinct collections from the 452-question storage-study cohort (Table 7; Figure 12). We identify duplicate collections by their source graph and relation overlay, check source-file hashes, and weight each collection once. From verified feature caches, we reconstruct routing digests, source locators, the cover hierarchy, source-unit membership sets, and relation edges without model calls. We serialize these with pickle protocol 5, optionally using gzip level 6. Deserialization preserves the nodes, graph structure, source-unit membership sets, and initial scene in every case.

Cache sizes exclude canonical sources, input indexes and overlays, images, source-opening helpers, and logs. Original-file sizes count verified UTF-8 source files and trajectory JSON, excluding images. The figure expresses each collection's cache size as a percentage of its original-file size.

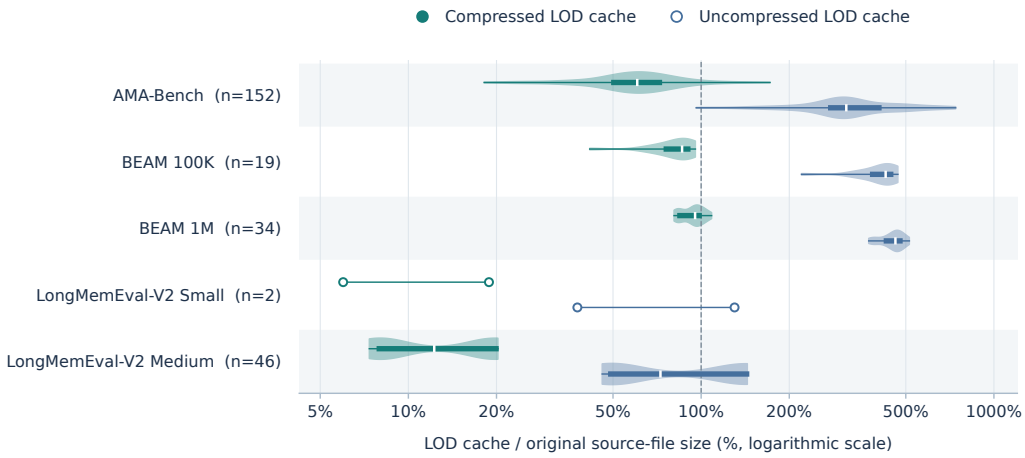


Figure 12: LOD cache storage as a percentage of original source-file size (excluding images). Violins use log-percentage kernel densities with Scott's bandwidth and independently normalized widths. The dashed line marks equal cache and source-file sizes (100%). White ticks mark medians; thick and thin segments span interquartile and full ranges. Groups with fewer than five collections show actual percentages. The axis is logarithmic; n counts distinct collections in the storage-study cohort.

Table 7: Average original-file and LOD cache sizes in MiB (2^{20} bytes) across the 253 distinct collections in the storage-study cohort. Arithmetic means weight collections equally. Original files comprise source text or trajectory JSON, excluding images.

Setting	Collections	Original files	Uncompressed LOD cache	Compressed LOD cache
BEAM 100K	19	0.586	2.383	0.476
BEAM 1M	34	4.457	20.232	4.180
LongMemEval-V2 Small	2	81.563	72.035	10.629
LongMemEval-V2 Medium	46	374.049	406.547	58.200
AMA-Bench	152	0.223	0.666	0.138