PERDOCBENCH: EVALUATING CROSS-SOURCE EVIDENCE INTEGRATION IN PERSONALIZED MEMORY SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalized memory agents are increasingly expected to integrate a user's conversations with private documents, yet most evaluations still rely on homogeneous chat streams that can reward benchmark-specific retrieval shortcuts. This makes progress difficult to interpret: strong performance may reflect adaptation to benchmark artifacts rather than robust use of heterogeneous personal evidence. We introduce **PerDocBench**, a cross-source benchmark for evaluating whether memory systems can connect conversational cues with document-grounded facts in persona-specific scholarly contexts. Unlike chat-only benchmarks, PerDocBench targets questions that require evidence to be identified, retained, and integrated across structurally different sources. Experiments with 16 representative memory methods under a shared backbone reveal a negative-utility pattern: averaged over three runs, 15 of 16 methods recover the requested fact less often than the backbone. These results suggest that current memory systems often degrade cross-source evidence integration in scholarly document-centered personalization. An anchor-reordering ablation yields a median L1 gain of 7.1% across systems and motivates *anchor-first, evidence-preserving memory construction* as a future design direction. The code is available at Anonymous GitHub.

1 INTRODUCTION

Personalized memory agents are expected to use several types of user context, including conversations, documents, profiles, and shared workspaces. Yet widely used benchmarks such as LoCoMo (Maharana et al., 2024), LongMemEval (Wu et al., 2024), and the PersonaMem family (Jiang et al., 2025) usually present supporting evidence in a single homogeneous chat stream. Strong scores in this setting may therefore reflect adaptation to benchmark-specific regularities rather than robust use of the heterogeneous context available to deployed systems.

Conversations often reveal a user's preferences or projects, while details needed later are scattered across much larger collections of notes, PDFs, reports, and email attachments stored across personal devices and cloud drives. For example, a chat may link "Jackfruit-Materials" to sustainable composites; a later query asks what enables "dynamic forms," but the answer—3D printing—appears only in a relevant paper. We call such document-linked exchanges *research-anchor conversations*: they provide a compact project- or topic-level cue that points to relevant documents. This creates a **cross-source cue–evidence asymmetry**: the conversation tells the system what to seek, while the documents contain the answer-bearing evidence. Memory systems must connect these sources while retaining the facts needed to answer.

We call the distribution and order of these cues and facts the context's *evidence-density structure*. The *evidence-density ratio* (EDR), a lexical proxy for evidence concentration, compares the density of distinct gold-answer content words in the research-anchor conversations with that in the document corpus. By this measure, the conversations are $6.5\times$ denser than the corpus (Appendix B.3). The compact evidence packaging used by chat-only benchmarks can hide this difficulty. To test sensitivity to such packaging, we add plausible but irrelevant details without changing the question, gold answer, or supporting fact. This answer-preserving rewrite turns up to 63.6% of originally correct predictions into incorrect ones (Figure 1(a), §3). We call this presentation sensitivity **benchmark hackability**.

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.

Table 1: Benchmark-design axes targeted by PerDocBench. ● satisfies; ◐ one source / partial; ○ does not. **Het. ev.**: chat and a different-format evidence store. **X-source**: question demands reasoning across source types. **Per.-disj.**: persona-disjoint evidence pool. **Real-doc**: real (non-templated) documents. **Q-type**: diversity beyond factual lookup. **#Users** counts personas / dialogue contexts; **#Q** counts released questions. "–" indicates not disclosed.

Benchmark	Het. ev.	X-source	Per.-disj.	Real-doc	Q-type	#Users	#Q
LoCoMo (Maharana et al., 2024)	○	○	○	○	◐	10	1,540
LongMemEval-S* (Wu et al., 2024; Hu et al., 2025)	○	○	○	○	◐	5‡	300
PersonaMem (Jiang et al., 2025)	○	○	◐	○	◐	180	∼6,000
PersonaConvBench (Li et al., 2025)	○	○	◐	◐	○	3,878	111,239
AlpsBench (Xiao et al., 2026)	○	○	◐	◐	◐	2,500	9,942
LaMP (Salemi et al., 2024)	○	○	◐	◐	◐	–	97,446
ATM-Bench (Mei et al., 2026)	●	◐	●	●	◐	1	1,044
BESPOKE (Kim et al., 2025)	●	◐	●	●	○	30	150
PerDocBench (ours)	●	●	●	●	●	10	266

‡LongMemEval-S* is the MemoryAgentBench reformulation of LongMemEval-S (Hu et al., 2025).

Among the benchmarks compared in Table 1, none covers all five properties. We introduce **PerDocBench**, which pairs a short conversation about a project or topic with a much larger, persona-specific collection of real papers. This mirrors a common use case: the conversation tells the system what the user is referring to, but the facts needed to answer the question are buried in the documents. The system must connect the conversation to the right papers, find the relevant facts, and keep those facts intact while building and searching its memory. Current memory methods struggle even though the full input fits within the language model's context window: 15 of 16 perform worse than the same model without an added memory method (Figure 1(b)). Analysis of one corpus-first run associates answer-string matches in the generator input with higher L1 accuracy, while individual cases reveal both altered memory content and incorrect use of retrieved evidence (§5.4). A simple reordering also helps: placing the conversation before the documents yields a median L1 gain of 7.1% across all 17 systems, with gains for 15 (§5.3). These results motivate *anchor-first, evidence-preserving memory construction*: present the conversational cue before processing the documents, and use summaries or other memory structures to locate original evidence rather than replace it. Our contributions follow:

C1. We show that compact chat-local evidence packaging is hackable: answer-preserving, density-reducing distractor rewrites flip up to 63.6% of correct memory-agent answers without changing the question or gold answer (§3).

C2. We construct **PerDocBench**, a cross-source personalized benchmark with 10 scholar personas, 97 real PDFs, and 266 QA pairs over four question types (§4). On three-run mean L1, 15 of 16 memory methods score below their shared OAI-4o-mini backbone (§5.2).

C3. Anchor-first reordering yields a median L1 gain of 7.1% over all 17 systems, with gains for 15 (§5.3). Case analyses identify failures in both memory content and answer generation and motivate *anchor-first, evidence-preserving memory construction* (§5.4).

2 Related Work

We compare the source formats used in personalized memory benchmarks and the representations maintained by memory systems. Appendix A provides the extended survey.

2.1 Personalized Memory Benchmarks

Personalized memory benchmarks differ primarily in the input format they assume. LoCoMo (Maharana et al., 2024), LongMemEval (Wu et al., 2024), the PersonaMem family (Jiang et al., 2025; Li et al., 2025), and AlpsBench (Xiao et al., 2026) present personalization through a single chat or interaction stream, with personalized facts strewn through the dialogue; since the supporting fact appears as a compact utterance, a method that retrieves the right window typically reproduces the right answer, and our distractor experiment (§3) shows that this surface is hackable. A second line, including PersonaFeedback (Tao et al., 2025), PersonaLens (Zhao et al., 2025), and LaMP (Salemi

(a) Density-rewriting one LoCoMo evidence utterance.

Original (compact, high-density evidence).
"**I went to a LGBTQ support group yesterday** and it was so powerful."

⇓ *answer-preserving density rewrite*

Variant (low-density, distractor-heavy).
"Oh, **I finally made it to that LGBTQ support group**, the one I'd been meaning to check out. It was at the community center downtown, and I was a bit nervous beforehand. Once inside, everyone was welcoming; I even grabbed a coffee from the cafe next door first. The rest of the day was scattered errands and messages; looking back, **all of that was yesterday**, and the session was incredibly powerful."

(b) On PerDocBench, 15 of 16 methods score below the OAI-4o-mini backbone.

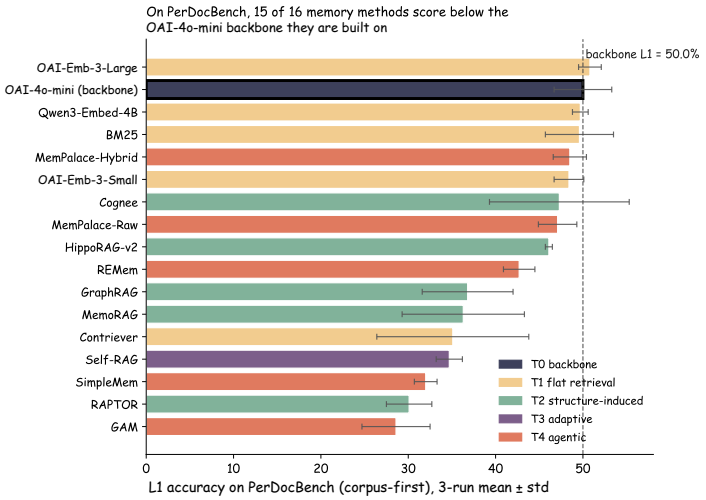


Figure 1: Hackability example and PerDocBench collapse. **(a)** An answer-preserving rewrite example; **blue** marks the preserved answer-supporting fact and gray marks plausible-but-irrelevant distractor detail. **(b)** PerDocBench results under the corpus-first condition; methods are sorted by L1 and colored by capability tier (T0 backbone, T1 flat retrieval, T2 structure-induced, T3 adaptive, T4 stateful agentic).

et al., 2024), fixes the personalization signal as an explicit profile block, testing whether a model can adapt to a given persona but not multi-source integration.

A third line broadens the evidence sources: ATM-Bench (Mei et al., 2026) mixes images, videos, and emails, BESPOKE (Kim et al., 2025) pairs chat with search histories, FingerTip-20K (Yang et al., 2025) adds GUI episodes, and PDR-Bench (Liang et al., 2025), the closest to our scholarly setting, targets long-form deep-research article generation. These benchmarks broaden the evidence carrier, but their sources need not play complementary routing and answering roles. Rather than adding modalities, PerDocBench binds a dense free-form chat anchor to a much larger, sparser persona-specific document corpus. The conversation identifies what to seek, whereas the corpus supplies the answer; each item is designed so that neither source suffices in isolation (§4). Appendix A compares all benchmarks by axis.

2.2 Memory Baseline Taxonomy

We organize memory baselines along a five-tier capability taxonomy according to what they write, compress, or preserve before answering, since this is where specific evidence can be lost. **T0** uses an in-context LLM, which answers from the supplied evidence without an external memory module. **T1** performs flat chunk retrieval: BM25 (Robertson & Zaragoza, 2009) ranks lexical matches, while Contriever (Izacard et al., 2021), OAI-Emb-3, and Qwen3-Embedding-4B (Zhang et al., 2025) strengthen retrieval through dense chunk–query embeddings. **T2** induces corpus structure before retrieval, from RAPTOR's summary tree (Sarthi et al., 2024) and graph-based routing in GraphRAG (Edge et al., 2024) and HippoRAG-v2 (Gutiérrez et al., 2025) to MemoRAG's clue-guided reformulation (Qian et al., 2025) and Cognee's hybrid graph–vector store. **T3** makes retrieval adaptive through Self-RAG's self-reflection signals (Asai et al., 2024), and **T4** maintains persistent agentic memory stores as in MemPalace, REMem (Shu et al., 2026), GAM (Yan et al., 2025), and SimpleMem (Liu et al., 2026). A shared backbone supports comparison across these memory designs.

3 Motivation: Evidence Density in Memory Evaluation

Document-centered memory often requires finding a small piece of evidence in a large collection of documents. In chat-based benchmarks, by contrast, the supporting fact is often stated in a short utterance. Before introducing PerDocBench, we therefore test a basic question: does memory

performance drop when the same evidence is presented less densely? On LongMemEval-S* and LoCoMo, we keep each question, gold answer, and supporting fact unchanged while making the supporting utterance less compact. If a previously correct prediction becomes incorrect, the system is sensitive to evidence density even though the answer remains supported. We call this sensitivity to evidence density *benchmark hackability*.

Controlled answer-preserving stress test. For each evaluated example, a separate LLM rewrites the supporting-evidence utterance to reduce its *evidence density* by inserting plausible but irrelevant side details, while leaving the question and gold answer unchanged and preserving support for the answer. We retain only variants that the rewriter's self-check identifies as still supporting the gold answer. We evaluate two representative methods at opposite ends of the memory-construction spectrum, **HippoRAG-v2** and **MemPalace-Raw**, on 69 LongMemEval-S* and 300 LoCoMo questions ($\approx$ 19%–23% of each benchmark). For *exact match* (EM) and *ROUGE-L F1*, we binarize scores at τ=0.5 and report the *flip ratio* $\rho_m = N_{R\to W}(m)/N_R(m)$: the fraction of originally correct predictions that become incorrect after rewriting. Appendix C.1 provides the full rewriter protocol, sample-selection rules, formal definitions, and method and metric rationale.

Illustrative rewrite. Figure 1(a) shows a LoCoMo example. The original utterance places an event beside its temporal cue, "yesterday." The rewrite inserts details about location, mood, coffee, and errands between these cues while preserving support for the same answer, "*7 May 2023*." This changes how densely the supporting evidence is presented, while leaving the question and its gold answer unchanged.

Table 2: Answer-preserving rewrite sensitivity. Correct-to-incorrect flip ratio ρ_m (%); higher is worse. R-L denotes ROUGE-L F1.

	LME-S*		LoCoMo	
Method	**EM**	**R-L**	**EM**	**R-L**
HippoRAG-v2	13.8	15.0	35.5	26.5
MemPalace-Raw	28.6	36.4	**63.6**	50.8

Sensitivity to evidence packaging. Table 2 reports the flip ratio for each method, metric, and dataset. Although all retained rewrites pass the answer-support self-check, EM flips occur for 13.8%–63.6% of originally correct predictions, while ROUGE-L F1 flips occur for 15.0%–50.8%. The largest degradation occurs for MemPalace-Raw on LoCoMo, where more than half of the originally correct predictions become incorrect under both metrics; all eight evaluated method–metric–dataset combinations exhibit nonzero degradation. The evaluated benchmark–method pairs are therefore substantially sensitive to evidence packaging: under our operational definition, **the sampled benchmark slices are partially hackable**, with answer-preserving rewrites overturning up to 63.6% of previously correct predictions. This diagnosis concerns score sensitivity in the tested methods and questions; it does not imply a general inability to reason robustly.

This sensitivity motivates a cross-source evaluation in which the conversation provides a cue and a separate document collection contains the answer-bearing evidence. PerDocBench implements this design and requires every question to connect the two sources (§4).

4 PerDocBench Construction

PerDocBench separates compact conversational routing cues from answer-bearing document evidence, instantiating the five design axes in Table 1. Each persona pairs a real-document library with an anchor-bearing conversation, and every final question requires evidence binding across them. Figure 2 shows a mechanical-engineering example: an interest in sustainable composites retrieves a personal library; a construction-only property graph seeds a document fact; and a conversational anchor turns that fact into a cross-source question. The graph is never shown to evaluated systems. Thus the conversation supplies the routing cue, while the corpus supplies answer evidence. Appendix B gives implementation details, hyperparameters, and full counts.

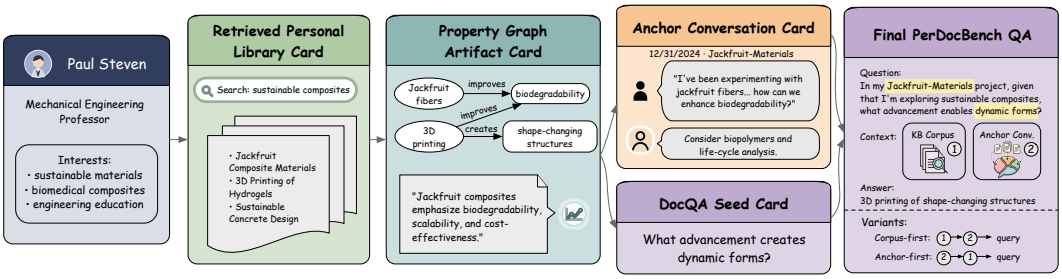


Figure 2: Pipeline for constructing PerDocBench. A persona profile is expanded into a persona-specific real-paper library, parsed into a property graph used as a construction artifact, and paired with an anchor conversation card. The final QA item binds both sources: the project codename and topical cue come from the conversation, while the answer is grounded in the document corpus.

4.1 Persona-Specific Document Libraries and Graph Scaffolds

Each evidence library follows a scholar-persona profile rather than a generic topic. OAI-5.4-mini maps the profile to taxonomy tags that seed Semantic Scholar retrieval. We retain each paper's best available form—parsed Markdown, PDF, or title-plus-abstract—in a persona-disjoint library.

Persona-specific libraries approximate coherent scholarly collections rather than a benchmark-wide topic pool, while disjointness limits cross-persona leakage. Raw paper collections alone do not provide a controlled basis for sampling questions across multiple retrieval and reasoning scales. For question design, we therefore transform each persona library into a property graph over paper chunks, typed entities and relations, together with source links. The graph normalizes repeated entity mentions across papers and exposes cross-paper paths, providing structured seeds for questions that go beyond single-passage lookup. It remains an internal construction scaffold: evaluated systems receive the same heterogeneous context—conversation plus corpus—and never the graph itself. The graph broadens question coverage without giving evaluated systems additional evidence.

4.2 Cross-Source Question and Anchor Construction

Question construction proceeds in three linked stages. First, the graph seeds four document-QA families that target complementary retrieval and reasoning scales: corpus-level browsing, section-level retrieval, single-fact technical detail, and multihop reasoning. Second, we generate time-stratified anchor conversations whose project codenames and topical cues refer indirectly to papers in the persona library. These anchors deliberately omit paper titles, preventing direct title matching from substituting for conversation-to-document association. Third, we match an anchor-conversation stem with a document-QA seed and rewrite the pair as a natural user query.

In Figure 2, for example, the query mentions the "Jackfruit-Materials" project introduced in conversation and asks about "dynamic forms," while the document corpus supplies the answer-bearing fact that "3D printing creates shape-changing structures." Each item therefore targets **cross-source evidence binding** rather than isolated-source recall: the conversation resolves the project reference, whereas the document corpus supplies the answer-bearing fact.

4.3 Grounding and Shortcut-Resistance Validation

Generative assembly can produce questions that appear cross-source while remaining answerable through leakage or shallow lexical cues. We therefore combine deterministic filters with an LLM-based semantic assessment. The filters screen for unsupported questions, weak anchor–question coupling, degenerate inventory phrasing, and answer leakage; the semantic assessment scores evidence support, answerability, context necessity, and shortcut resistance against retrieved evidence snippets. Under the shared cross-dataset assessment protocol, Appendix B reports higher grounding, lower duplication, and a higher strict-pass rate for PerDocBench than for LoCoMo and LongMemEval-S*. Quality control checks answer support and the intended *division of labor between sources*, going beyond whether a question merely mentions both sources.

Released benchmark. After final filtering, the 292 assembled candidates yield 266 released question–answer pairs across ten personas, with 20–30 questions per persona. The set contains 110 Technical Detail, 66 Section Retrieval, 48 Multihop Reasoning, and 42 Document Browsing questions. This uneven distribution motivates reporting both overall accuracy and separate scores for each question type in Section 5. The persona libraries index 677 papers, with 26–133 papers per persona; 97 PDFs are parsed for graph construction, a count distinct from the full metadata library. Per-persona memory contexts span 21k–72k whitespace-delimited tokens, averaging approximately 44k. Appendix B.2 reports the per-persona construction counts and document coverage.

5 EXPERIMENTS

To assess cross-source performance across memory designs, we evaluate 16 memory systems under a shared answering backbone and a common two-stage interface. Beyond overall performance, we examine the effect of anchor-first ordering and the relationship between intermediate evidence and final answers through three research questions:

- **RQ1:** Does PerDocBench challenge current memory systems' cross-source integration?
- **RQ2:** Can simple reordering of materials mitigate its evidence-density challenge?
- **RQ3:** What do intermediate outputs and failure cases reveal about evidence use?

5.1 SETUP

Systems and protocol. We compare 16 memory methods with a shared in-context backbone and organize them with the T0–T4 capability taxonomy. T0 is the OAI-4o-mini backbone. T1 comprises the flat retrievers BM25 (Robertson & Zaragoza, 2009), Contriever (Izacard et al., 2021), OAI-Emb-3-Small, OAI-Emb-3-Large, and Qwen3-Embedding-4B (Zhang et al., 2025). T2 comprises the structure-induced retrievers RAPTOR (Sarthi et al., 2024), GraphRAG (Edge et al., 2024), MemoRAG (Qian et al., 2025), Cognee, and HippoRAG-v2 (Gutiérrez et al., 2025). T3 contains Self-RAG (Asai et al., 2024). T4 contains MemPalace-Hybrid, MemPalace-Raw, REMem (Shu et al., 2026), GAM (Yan et al., 2025), and SimpleMem (Liu et al., 2026). Appendix C documents each method's construction and query interface.

Implementation and metrics. All systems answer with OAI-4o-mini, and every per-persona context fits within its window without truncation (Appendix B.2). Thus, context-window overflow does not explain the memory methods' underperformance relative to the backbone. We use 4,096-token chunks and cap generation at 256 tokens. Qwen3-Embedding-4B, Contriever, HippoRAG-v2, and MemoRAG run locally; other model inference is served through APIs. OAI-4o judges each response under fixed prompts. **L1** credits responses that contain the requested gold fact or an unambiguous equivalent, whereas **L2 strict** additionally requires the response to be precise, task-faithful, and free of unsupported or conflicting content. Scores are reported as percentages. We further report L1 for Technical Detail (TD), Section Retrieval (SR), Document Browsing (DB), and Multihop Reasoning (MR). Appendices C.3 and C.5 provide metric definitions, lexical metrics, timing, and full per-type results. Results characterize the implementations under this shared protocol; method-specific adaptations are detailed in Appendix C.2.

Judge validation. Two author annotators independently re-labeled a stratified 150-item subset, blind to machine verdicts. It spans ten methods and all four question types, with boundary cases deliberately oversampled. Pooled human–judge agreement is 91.7% for L1 and 81.3% for L2; an alternative judge agrees on 91.3% and 88.0%, respectively. These figures describe the selected subset, with lower consistency for the strict criterion. Agreement alone does not establish that system rankings would remain unchanged under a different judge. Appendix C.4 reports the sampling protocol, inter-annotator agreement, and confidence intervals.

Controlled source order. We test two anchor placements. **Corpus-first** presents the persona corpus before the anchor-bearing conversation, whereas **anchor-first** reverses these blocks. The query follows both. This intervention holds content, length, and EDR fixed, changing only the order component of the evidence-density structure. We evaluate corpus-first ordering over three runs and pair one of these with an anchor-first run.

Table 3: Corpus-first results on PerDocBench. Accuracy entries are mean ± sample standard deviation over three runs and are reported in percent. TD, SR, DB, and MR denote Technical Detail, Section Retrieval, Document Browsing, and Multihop Reasoning. **Bold** and underlined entries mark the best and second-best non-backbone results within each metric. Mem is mean memory-construction time per persona and Q is mean per-query latency, both in seconds; row colors indicate capability tier. Appendix C.5 reports full single-run metrics and the anchor-first ablation.

Method	Tier	L1 (%)	L2 Strict (%)	TD-L1	SR-L1	DB-L1	MR-L1	Mem (s)	Q (s)
OAI-4o-mini (backbone)	T0	50.0 ± 3.3	28.7 ± 4.1	56.2 ± 3.3	53.0 ± 2.7	32.8 ± 4.6	48.3 ± 1.4	0.00	3.84
BM25	T1	49.6 ± 3.9	29.2 ± 1.1	**59.7** ± 5.8	53.4 ± 1.0	27.4 ± 11.3	43.0 ± 4.8	0.00	4.69
OAI-Emb-3-Large	T1	**50.8** ± 1.3	29.0 ± 1.3	56.4 ± 1.6	**62.1** ± 1.3	23.5 ± 8.5	**47.6** ± 0.2	1.12	3.95
OAI-Emb-3-Small	T1	48.4 ± 1.7	28.1 ± 0.4	58.1 ± 0.7	55.0 ± 3.6	24.3 ± 5.8	41.5 ± 1.6	1.02	3.97
Qwen3-Embed-4B	T1	49.7 ± 0.9	27.9 ± 2.8	54.6 ± 2.4	59.8 ± 2.0	25.0 ± 1.0	47.4 ± 3.1	0.04	3.89
Contriever	T1	35.1 ± 8.7	19.9 ± 4.6	46.5 ± 3.1	33.9 ± 16.6	18.1 ± 7.0	30.1 ± 8.2	0.01	3.80
Cognee	T2	47.3 ± 8.0	30.9 ± 5.0	54.6 ± 8.7	48.4 ± 5.2	36.7 ± 9.1	40.7 ± 11.8	83.87	12.82
HippoRAG-v2	T2	46.1 ± 0.4	**32.0** ± 1.3	56.4 ± 1.6	47.6 ± 0.8	26.6 ± 1.7	41.0 ± 0.6	20.31	13.77
GraphRAG	T2	36.8 ± 5.2	20.4 ± 1.7	44.1 ± 8.3	41.5 ± 9.7	19.5 ± 1.6	31.9 ± 0.5	22.72	28.58
MemoRAG	T2	36.3 ± 7.0	23.3 ± 4.5	44.9 ± 6.0	36.4 ± 1.3	15.8 ± 19.3	36.5 ± 4.5	1.68	10.21
RAPTOR	T2	30.1 ± 2.6	18.6 ± 1.5	35.7 ± 2.9	28.2 ± 1.8	21.1 ± 0.3	29.2 ± 3.6	264.98	**1.43**
Self-RAG	T3	34.7 ± 1.5	19.5 ± 2.6	42.7 ± 4.0	37.3 ± 4.7	10.2 ± 1.5	35.8 ± 3.3	352.29	17.83
MemPalace-Hybrid	T4	48.5 ± 1.9	29.8 ± 0.2	57.6 ± 2.1	55.4 ± 1.9	18.9 ± 8.4	46.0 ± 5.6	0.51	3.89
MemPalace-Raw	T4	47.1 ± 2.2	29.3 ± 1.3	58.9 ± 2.6	51.5 ± 2.6	22.7 ± 3.1	38.9 ± 3.1	0.51	3.82
REMem	T4	42.7 ± 1.8	27.6 ± 0.8	52.2 ± 2.0	41.4 ± 4.8	28.1 ± 0.4	38.2 ± 4.3	0.09	9.54
SimpleMem	T4	32.0 ± 1.3	21.3 ± 1.5	33.0 ± 1.8	28.7 ± 0.1	**37.5** ± 0.5	29.8 ± 4.2	30.67	21.73
GAM	T4	28.6 ± 3.9	12.8 ± 1.9	35.4 ± 3.1	26.2 ± 0.9	21.7 ± 14.7	24.3 ± 4.8	183.50	20.16

5.2 Cross-Source Performance on PerDocBench (RQ1)

Comparison with the shared backbone. Table 3 reports corpus-first results averaged over three independent runs. On L1, 15 of 16 memory methods score below the shared OAI-4o-mini backbone. OAI-Emb-3-Large is the sole exception, but its 0.8% L1 advantage is smaller than the variation observed across runs and does not provide clear evidence of an improvement. L2 strict is more mixed, with six memory methods above the backbone. Across the 16 memory methods, mean L1 is 41.5% and median L1 is 44.4%, compared with 50.0% for the backbone; seven methods score below 40.0%. **PerDocBench therefore poses a genuine cross-source challenge: adding a memory layer usually lowers L1 accuracy even when the answering backbone is fixed.**

Analysis by memory design. Performance aligns more closely with evidence preservation than with capability tier. Five of the six methods that outperform the backbone on L2 nevertheless score lower on L1. HippoRAG-v2, for example, scores 32.0% on L2 but 46.1% on L1; the backbone scores 28.7% and 50.0% on the same metrics. The leading L2 methods retain query-time access to local or verbatim evidence, whereas systems that rewrite, compress, or filter evidence risk losing document-specific details. Access to searchable evidence is therefore more informative than capability tier for interpreting these results.

Analysis by question type. No method dominates across question types. Different memory methods lead Technical Detail, Section Retrieval, and Document Browsing, while the backbone remains strongest on Multihop Reasoning. All 16 memory methods score below the backbone on Multihop Reasoning L1, averaging 37.6% compared with 48.3% for the backbone. For Document Browsing, 14 methods likewise score below the backbone's 32.8%. On L2 strict, Multihop Reasoning has the largest mean deficit and Document Browsing degrades for most methods. These results reveal complementary weaknesses across memory designs; the full matrix appears in Appendix C.5.

Analysis by computational cost. Measured construction time spans five orders of magnitude, yet higher runtime is not associated with better accuracy under the evaluated execution setup. With an average of 26.6 queries per persona, we define the amortized time as $C = T_{\text{mem}} + 26.6T_{\text{query}}$. By this measure, 13 of the 16 memory methods take longer per persona than the backbone while also yielding lower mean L1. Mean L1 correlates negatively with C: Spearman's ρ is -0.59, with p=0.014. The association is stronger for construction time alone: ρ is -0.65, with p=0.004. The corresponding L2 associations are weaker and not statistically significant. Figure 3 visualizes this relationship, and Appendix C.5 reports the full timings. **More computation is not associated with better answers.**



Figure 3: Compute–accuracy relationship. Three-run mean L1 versus log construction time under corpus-first ordering. Times below 10^{-3} s are clipped at that threshold before applying $\log_{10}$. Bubble area denotes query time, color denotes capability tier, and the dashed line marks the backbone.

Neither architectural complexity nor additional computation resolves the observed cross-source challenge. We next test whether presenting the same materials in a different order improves performance.

5.3 Source Order in the Evidence-Density Structure (RQ2)

Effect of source order. To isolate source-order sensitivity, we compare one corpus-first run with one anchor-first run for every system while holding content, length, and EDR fixed. Placing the conversation first yields a median L1 gain of 7.1% across the paired system differences, and 15 of 17 systems improve. However, every memory method still trails the anchor-first backbone. **Source order matters, but reordering is only a partial remedy.** Table 15 in Appendix C.6 reports all per-method and per-metric differences.

What the reordering reveals. With corpus-first ordering, a system may choose what to retrieve or compress before seeing the cue that identifies the relevant detail. Placing the conversation first lets that cue guide processing, but cannot restore evidence already discarded by a lossy memory store. The backbone also improves, suggesting that general long-context position effects contribute. The ablation therefore establishes sensitivity to source order, not a single underlying cause.

Consistency across backbones. We also test source ordering across five full-context backbones on the same 266 questions, holding data, prompts, judge, decoding settings, and seed fixed. All five improve under anchor-first ordering, with L1 gains of 2.6%–12.0%. The direction of the source-order effect therefore extends beyond the shared backbone. This comparison uses no retrieval or memory module, so it does not establish how memory methods would perform with other backbones. Appendix C.7 reports the full results.

5.4 Answer Evidence and Failure Cases (RQ3)

We relate answer correctness to answer-string matches in the generator input, then inspect a case to distinguish errors in memory content from errors during generation.

Answer-string matches and correctness. Using one corpus-first run, we analyze 3,974 method–query instances from the 15 methods with alignable intermediate outputs. We test whether any normalized gold-answer string occurs in the recorded generator input. L1 accuracy is 63.3% among the 772 matched instances, compared with 39.3% among the 3,202 unmatched instances. Answer-string matching is therefore associated with correctness, but does not establish whether the input contains sufficient evidence: it misses paraphrases, and a match to one gold string need not cover a multi-item answer. Of the 2,228 incorrect outputs, 1,945 (87.3%) have no match. This identifies a common feature of the observed errors, not the stage or cause of failure. Figure 4 summarizes the outcomes; Appendix C.9 gives the full counts and matching rule.

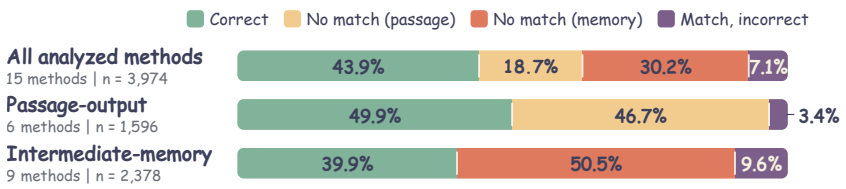


Figure 4: Answer outcomes in one corpus-first run. Correct includes all L1-correct outputs, regardless of answer-string matching. Incorrect outputs are split into no match in passage-output or intermediate-memory pipelines, and match but incorrect. A match tests answer wording, not evidence sufficiency. Table 21 in Appendix C.9 reports counts and per-method values.

Table 4: Representative anchor-question failure under corpus-first ordering. Outputs are reproduced from the judge log. **Green** marks the gold span and **red** an incorrect substitution. Italic parentheticals identify *prior snap* (substitution with a familiar default), *paraphrase drift* (loss of a distinguishing qualifier), and *summary erasure* (omission of the answer-bearing detail). Table 17 in Appendix C.8 provides the extended comparison.

Method	Tier	Output (verbatim)	Failure mode
OAI-4o-mini	T0	"…**cradle-to-construction site**, including the product stage…"	**correct**
HippoRAG-v2	T2	"**Cradle-to-construction site** system boundary."	**correct**
Cognee	T2	"…the **cradle-to-construction site** approach."	**correct**
GraphRAG	T2	"…from **cradle to grave**, covering all life cycle stages…"	wrong detail (*prior snap*)
Self-RAG	T3	"A **cradle-to-grave** LCA approach is defined…"	wrong detail (*prior snap*)
Contriever	T1	"…from **raw material extraction… to end-of-life**…"	wrong detail (*paraphrase drift*)
RAPTOR	T2	"**not explicitly defined** in the provided context."	non-answer (*summary erasure*)

Representative case. Table 4 illustrates these stages with one Technical Detail question from the Eco-Design persona. The question asks for the system boundary used in the persona's mass-timber LCA study; its gold answer, **cradle-to-construction site**, differs from the more familiar *cradle-to-grave* scope. Four of the seven systems shown answer incorrectly, and GraphRAG and Self-RAG both substitute the familiar *cradle-to-grave* scope for the document-specific answer.

Analysis of recurring failure patterns. Although GraphRAG and Self-RAG both answer *cradle-to-grave*, their errors occur at different stages. GraphRAG's constructed memory has already replaced the document's boundary with *cradle-to-grave*, so its prior snap occurs during memory construction. Self-RAG retrieves four passages containing the complete gold span but makes the same replacement during generation. Every other system shown that retains the complete span answers this question correctly. Appendix C.8 reports additional cases.

Design implications. The case illustrates why answer wording must be interpreted with its qualifier and local context. An intermediate memory can replace a document-specific fact, while a generator can also misuse a passage that retains it. Together with the source-order results, these observations motivate *anchor-first, evidence-preserving memory construction*: present routing cues before compression and use summaries or graphs to index searchable source passages.

6 Conclusion

PerDocBench evaluates whether personalized memory systems can connect conversational cues with evidence in persona-specific scholarly libraries. Motivated by answer-preserving rewrites that expose presentation sensitivity in chat-based evaluations, it requires systems to preserve and integrate information across sources. Under corpus-first ordering, 15 of 16 memory methods underperform their shared OAI-4o-mini backbone on three-run mean L1. Anchor-first reordering yields a median L1 gain of 7.1% across paired system differences. Answer-string matching is associated with correctness, while inspected cases show errors both in intermediate memories and during answer generation. Within this protocol, these findings motivate presenting routing cues before compression and retaining searchable source passages indexed by summaries or graphs.

AI USE STATEMENT

We used generative AI tools for writing assistance and language editing, literature retrieval and discovery, and synthetic data generation. The data construction and validation procedures are described in Section 4 and Appendix B. The authors take responsibility for the final content of this work.

ETHICS STATEMENT

PerDocBench uses public scholarly sources and benchmark persona contexts rather than private user records. Source access and redistribution follow the applicable terms: the intended release includes derived instances, metadata, prompts, source links, and code, while full-text documents are included only where redistribution is permitted. Generated benchmark content is screened for unsupported content, answer leakage, unrelated personal identifiers, and unsafe content (Appendix B.10). The judge-validation study uses annotations from two authors, with no external participants (Appendix C.4). The benchmark is intended for research evaluation and does not establish readiness for deployment on private user data.

REPRODUCIBILITY STATEMENT

Section 5.1 specifies the shared evaluation protocol. Appendices B, C, and C.3 describe data construction, method adaptations, and metrics; Appendix D supplies the prompts, and Appendix C.5 reports detailed results. The anonymous code repository linked in the abstract provides the accompanying implementation.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

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. *arXiv preprint arXiv:2404.16130*, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*, 2025.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. *arXiv preprint arXiv:2507.05257*, 2025.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J Taylor, and Dan Roth. Know me, respond to me: Benchmarking llms for dynamic user profiling and personalized responses at scale. *arXiv preprint arXiv:2504.14225*, 2025.

Hyunseo Kim, Sangam Lee, Kwangwook Seo, and Dongha Lee. Bespoke: Benchmark for search-augmented large language model personalization via diagnostic feedback. *arXiv preprint arXiv:2509.21106*, 2025.

Li Li, Peilin Cai, Ryan A Rossi, Franck Dernoncourt, Branislav Kveton, Junda Wu, Tong Yu, Linxin Song, Tiankai Yang, Yuehan Qin, et al. A personalized conversational benchmark: Towards simulating personalized conversations. *arXiv preprint arXiv:2505.14106*, 2025.

Yuan Liang, Jiaxian Li, Yuqing Wang, Piaohong Wang, Motong Tian, Pai Liu, Shuofei Qiao, Runnan Fang, He Zhu, Ge Zhang, et al. Towards personalized deep research: Benchmarks and evaluations. *arXiv preprint arXiv:2509.25106*, 2025.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for llm agents. *arXiv preprint arXiv:2601.02553*, 2026.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

Jingbiao Mei, Jinghong Chen, Guangyu Yang, Xinyu Hou, Margaret Li, and Bill Byrne. According to me: Long-term personalized referential memory qa. *arXiv preprint arXiv:2603.01990*, 2026.

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*, pp. 2366–2377, 2025.

Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc, 2009.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, 2024.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pp. 32628–32649, 2024.

Yiheng Shu, Saisri Padmaja Jonnalagedda, Xiang Gao, Bernal Jiménez Gutiérrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. Remem: Reasoning with episodic memory in language agent. *arXiv preprint arXiv:2602.13530*, 2026.

Meiling Tao, Chenghao Zhu, Dongyi Ding, Tiannan Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. Personafeedback: A large-scale human-annotated benchmark for personalization. *arXiv preprint arXiv:2506.12915*, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

Jianfei Xiao, Xiang Yu, Chengbing Wang, Wuqiang Zheng, Xinyu Lin, Kaining Liu, Hongxun Ding, Yang Zhang, Wenjie Wang, Fuli Feng, et al. Alpsbench: An llm personalization benchmark for real-dialogue memorization and preference alignment. *arXiv preprint arXiv:2603.26680*, 2026.

BY Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. General agentic memory via deep research. *arXiv preprint arXiv:2511.18423*, 2025.

Qinglong Yang, Haoming Li, Haotian Zhao, Xiaokai Yan, Jingtao Ding, Fengli Xu, and Yong Li. Fingertip 20k: A benchmark for proactive and personalized mobile llm agents. *arXiv preprint arXiv:2507.21071*, 2025.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Zheng Zhao, Clara Vania, Subhradeep Kayal, Naila Khan, Shay B Cohen, and Emine Yilmaz. Personalens: A benchmark for personalization evaluation in conversational ai assistants. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18023–18055, 2025.

Appendix Contents

A Extended Survey of Related Work 13

A.1 Personalized Memory Benchmarks 13

A.2 Memory-Agent Baselines: Capability Taxonomy 15

B PerDocBench Data Construction Details 17

B.1 Persona Seed and Domain Coverage 17

B.2 Per-persona Detailed Statistics 19

B.3 Evidence-Density Ratio 20

B.4 Property-Graph Schema 21

B.5 Question-Generation Details 22

B.6 Anchor-Conversation Generation 22

B.7 Cross-Source Assembly and the F1–F6 Quality Filter 23

B.8 Cross-dataset Quality Assessment Details 23

B.9 Distractor-rewrite attack 24

B.10 Artifact Sources, Licenses, and Intended Use 24

C Supplemental Experiments and Results 25

C.1 Distractor-Rewrite Hackability Study 25

C.2 Per-Method Adaptation Notes 26

C.3 Metric Definitions 27

C.4 Judge Validation: Human and Cross-Model Agreement 29

C.5 Full Main and Ablation Tables 29

C.6 Full Per-Metric Delta Table 30

C.7 Long-Context Backbone Sensitivity 30

C.8 Extended Case Studies: Multihop and Document-Browsing 31

C.9 Answer-String Matching and Outcome Analysis 33

D Prompts 35

D.1 Dataset Construction Prompts 36

D.2 Anchor-Conversation and Cross-Source Prompts 38

D.3 LLM-Judge Prompts 40

D.4 Distractor-Rewrite Attack Prompt 41

E Limitations 42

A Extended Survey of Related Work

This appendix expands the related-work discussion of §2 along two axes. The first is an extended survey of personalized memory benchmarks, grouped by their evidence-container assumption in §A.1. The second is an extended survey of the memory-agent baselines we evaluate, organized by the T0–T4 capability taxonomy in §A.2. Throughout, we contrast each work with PerDocBench by the property it does or does not satisfy.

A.1 Personalized Memory Benchmarks

A.1.1 Conversation-as-context Benchmarks

The dominant evaluation surface in this line is a single homogeneous chat channel, possibly multi-session, in which personalized facts are dispersed across the dialogue. The carrier is uniform, and the personalization signal lives entirely in turn content.

LoCoMo (Maharana et al., 2024). 50 long multi-session synthetic dialogues (∼300 turns / 9K tokens each), produced by an LLM persona-conditioned generator. Tasks: single-hop, multi-hop, temporal, open-domain, adversarial. All evidence resides in one synthetic chat per pair; this is the prototype of the conversation-as-context shortcut surface and the dataset we attack with the distractor rewrite (§3). LoCoMo *does* use a per-persona temporal event graph at construction time, but the graph is hand-authored from persona templates to drive *dialogue* generation, not extracted from a per-persona document corpus to seed cross-source QA, so the graph-seeding axis we claim for PerDocBench does not collapse onto it.

LongMemEval and the LongMemEval-S* split (Wu et al., 2024; Hu et al., 2025). The original *LongMemEval* (Wu et al., 2024) ships 500 carefully written questions over long timestamped chat sessions and exposes three release sizes that share the same questions but differ in haystack length, of which *LongMemEval-S* (∼115K-token haystack, 500 questions, exactly *one* question per constructed history) is the smallest and is the variant most heavily used in subsequent memory-method papers (e.g., the Mem0 / A-Mem evaluations). The authors' own ablations already show that session decomposition, fact-augmented key expansion, and time-aware query rewriting lift accuracy—i.e., the benchmark's summary score is contestable by retrieval-time tricks rather than memory architecture. Hu et al. (2025) subsequently introduce **LongMemEval-S*** (the variant we adopt): they reformulate the LongMemEval-S chat history into *five long dialogues* (∼355K tokens each) carrying *300 questions* in total, so memory only has to be constructed once per dialogue rather than once per question, and the per-question cost of the memory-construction stage is amortised. Throughout this paper, all numbers we report on this benchmark—the distractor-rewrite flip ratios in §3 and the cross-dataset quality assessment in §4.3—are computed on the LongMemEval-S* split; we cite the parent paper (Wu et al., 2024) only in the original-source sense (chat data, question authoring) and the LongMemEval-S* paper (Hu et al., 2025) for the 5-dialogue / 300-question reformulation. The intro Table 1 #Users / #Q entry for this row therefore reads 5 / 300, not the 500 / 500 of the original LongMemEval-S subset.

PersonaMem family (Jiang et al., 2025). Implicit-persona long-context benchmark with ∼1k simulated users and up to 128K tokens of synthetic chat per user; evaluates whether the model infers latent preferences without an explicit profile. The benchmark emphasizes implicit user persona learning from conversation history rather than a profile block. The carrier remains a single synthetic chat per user, so the input format is still uniform.

PersonaConvBench (Li et al., 2025). Personalized multi-turn conversation grounded in real Reddit trajectories across 10 subreddits.

AlpsBench (Xiao et al., 2026). 2,500 real WildChat sequences with a four-stage memory lifecycle covering extraction, update, retrieval, and utilization.

A.1.2 PROFILE-AS-CONTEXT BENCHMARKS

This line factors out the dialogue and asks whether a model can adapt to an explicit profile block, typically a small set of attributes or a list of past interactions. Personalization is delivered through structured but single-source signals.

LaMP (Salemi et al., 2024). Seven personalized language tasks covering classification and generation, each with a per-user history of past items such as citations, tags, tweets, headlines, and ratings. The carrier is structured but single-source. LaMP remains the most widely cited personalization baseline for the profile-as-context regime.

PersonaFeedback (Tao et al., 2025). 8,298 human-annotated tuples for response selection, each containing a persona, query, and response pair. The benchmark evaluates persona alignment of *one* reply, not memory composition over a corpus. It is closer to reward modeling than memory-agent evaluation.

PersonaLens (Zhao et al., 2025). 1,500 semi-synthetic users over 111 task-oriented assistant tasks, with demographics drawn from PRISM and the remaining memory content generated by ANTH-3. The benchmark combines real demographics with synthetic memory, and the personalization signal is per-user demographics plus a small bag of tasks.

PDR-Bench (Liang et al., 2025). 25 authentic user profiles, 50 research tasks, 250 personalized queries; evaluates long-form deep-research articles via a personalization–quality–reliability rubric.

A.1.3 MULTI-SOURCE HETEROGENEOUS BENCHMARKS

This line widens the input modality. The supporting context is no longer a single chat, but the multiple channels are typically presented as a flat concatenation in a fixed schema, with multi-hop questions phrased *within* a stream rather than across structurally different containers.

ATM-Bench (Mei et al., 2026). $\sim$12K real personal memory items spanning images, video, and email over four years, with $\sim$2.25M tokens of context and $\sim$1.1K QA pairs. About 30% require multi-evidence composition. ATM-Bench is the strongest existing real-multimodal benchmark, but it differs in whether the Multihop Reasoning spans containers of very different information densities.

BESPOKE (Kim et al., 2025). 30 real users, 717 chat-history rows, 2,150 search-history rows, and 150 queries with per-query gold rubrics and diagnostic feedback. BESPOKE pioneers diagnostic-feedback evaluation, but the two text streams are parallel rather than structurally different.

FingerTip-20K (Yang et al., 2025). 21K real Android GUI episodes from 91 users. Out of the strict scope as it targets mobile-agent action personalization rather than textual memory QA, but included here for completeness because it is the largest real-multi-source personal-data benchmark we are aware of.

A.1.4 COMPARISON TABLE

Table 5 contrasts PerDocBench with the 12 benchmarks above along five axes that, to our knowledge, no prior benchmark table has compiled jointly: heterogeneous evidence container, cross-source reasoning required, persona-disjoint evidence pool, real-document grounding, and question-type diversity beyond factual lookup. Axes are chosen so that a satisfying entry cannot be added by a wrapper around a long-context model or by tuning a retrieval prompt. An axis therefore receives a satisfying entry only when the property is present in the data itself.

The pattern is consistent. Single-channel benchmarks fail at least the first three axes. Multi-source benchmarks improve on container and grounding but still pose Multihop Reasoning *within* a stream. PerDocBench is the first benchmark we are aware of for which all five axes are satisfied simultaneously, which is what makes the under-the-backbone collapse in §5.2 non-trivial.

Table 5: Comparison of personalized memory benchmarks along five PerDocBench-relevant axes. Filled circle (●) = satisfies; half-circle (◐) = partial / one source; open circle (○) = does not satisfy.

Benchmark	Het. ev.	X-source	Per.-disj.	Real-doc	Q-type
Conversation-as-context					
LoCoMo (Maharana et al., 2024)	○	○	○	○	◐
LongMemEval (Wu et al., 2024)	○	○	○	○	◐
PersonaMem (Jiang et al., 2025)	○	○	◐	○	◐
PersonaConvBench (Li et al., 2025)	○	○	◐	◐	○
AlpsBench (Xiao et al., 2026)	○	○	◐	◐	◐
Profile-as-context					
LaMP (Salemi et al., 2024)	○	○	◐	◐	◐
PersonaFeedback (Tao et al., 2025)	○	○	○	◐	○
PersonaLens (Zhao et al., 2025)	○	○	◐	○	◐
PDR-Bench (Liang et al., 2025)	◐	○	◐	●	○
Multi-source heterogeneous					
ATM-Bench (Mei et al., 2026)	●	◐	●	●	◐
BESPOKE (Kim et al., 2025)	●	◐	●	●	○
FingerTip-20K (Yang et al., 2025)	●	○	●	●	○
PerDocBench (ours)	●	●	●	●	●

A.2 Memory-Agent Baselines: Capability Taxonomy

We organize baselines using a five-tier capability taxonomy that characterizes how much external structure or persistent state the system maintains outside the language model. Each tier subsumes the previous tier's representational budget while introducing a new failure mode of its own. The ordering is not a quality ranking. Our experiments are designed so that a system's position in the taxonomy is informative on its own, independent of the embedding or LLM backbone it uses.

A.2.1 T0 — Closed-book / Long-context LLM

OAI-4o-mini. OAI-4o-mini is a small, cost-efficient multimodal chat model with a 128K-token context window. We use it as a in-context baseline: the persona's full evidence stream is concatenated into the prompt and the model answers without any retrieval or memory module. This isolates the contribution of the higher-capability tiers. There is no memory-writing step, and the query step reads the concatenated context.

A.2.2 T1 — Flat Retrieval over Chunks

T1 systems pre-split the corpus into chunks and rank them with a single scoring function. The corpus has no internal structure; only the scoring function differs.

BM25 (Robertson & Zaragoza, 2009). The canonical bag-of-words probabilistic retriever, scoring documents by IDF-weighted sums of saturating, length-normalized term frequency. It is robust to distribution shift and remains a competitive non-parametric baseline. The memory-write step indexes chunks, and the query step returns the top 10 by BM25 score, with a final OAI-4o-mini answer pass.

Contriever (Izacard et al., 2021). A bi-encoder dense retriever trained without labelled query–document pairs by contrastive learning over independent crops of the same document with in-batch and MoCo-style negatives. The unsupervised checkpoint matches BM25 zero-shot on BEIR. The

memory-write step embeds chunks, the query step returns the top 10 by cosine similarity, and the answer pass uses OAI-4o-mini.

OAI-Emb-3-Small/Large. OAI-Emb-3 models are two general-purpose embedders with native dimensionalities of 1,536 for the small model and 3,072 for the large model, trained with Matryoshka representation learning so truncated dimensions degrade gracefully. We include both to dissociate the cost–quality trade-off on our task. The memory-write step embeds chunks, the query step returns the top 10 by cosine similarity, and the answer pass uses OAI-4o-mini.

Qwen3-Embedding-4B (Zhang et al., 2025). A 4B-parameter open-weight embedding model built on the Qwen3 decoder LLM and trained via large-scale weakly-supervised contrastive pre-training, supervised fine-tuning, and model merging. Probes whether scaling the *embedder*, while keeping the retrieval mechanism flat, suffices for cross-document personalized reasoning.

A.2.3 T2 — Structure-Induced Retrieval

T2 systems build an offline structure over the corpus and route retrieval through that structure. Their inductive biases differ in *what* structure is induced.

RAPTOR (Sarthi et al., 2024). Embeds leaf chunks, soft-clusters them with Gaussian mixtures, and uses an LLM to summarize each cluster, recursing on the summaries to form a bottom-up tree of progressively abstract nodes. Retrieval ranges across all tree levels, so a single query can pull both granular evidence and document-level abstractions.

GraphRAG (Edge et al., 2024). Extracts an entity–relation knowledge graph from the corpus, partitions it into a hierarchy of communities via Leiden, and pre-computes an LLM *community report* for each. Global queries are answered by fanning out to a chosen hierarchy level and map-reducing partial answers; this is the canonical recipe for query-focused *global* sense-making.

HippoRAG-v2 (Gutiérrez et al., 2025). An LLM extracts an open knowledge graph of entities and triples, serving as the hippocampal "index," while passage embeddings supply neocortical context. A query is parsed into entities and mapped to seed nodes. Personalized PageRank then propagates relevance across the graph, and chunks are scored via aggregated PageRank mass over their entities. We use the OAI-Emb-3-Small variant in our main runs.

MemoRAG (Qian et al., 2025). MemoRAG pairs a light, long-range, KV-compressed "memory" model with a stronger answer model. The memory model digests the entire corpus into a compact global representation and, given a query, emits staged *clues* that approximate where the answer lives. Those clues are issued as expanded retrieval queries against an evidence retriever. The memory model thus acts as a globally aware *query reformulator*, not a final answerer.

Cognee. Cognee[1] ingests heterogeneous raw inputs spanning text, documents, audio, and images, then converts them into a persistent hybrid of knowledge graph and vector index via the Extract–Cognify–Load pipeline. *Extract* ingests and deduplicates with metadata. *Cognify* runs Pydantic-schema-guided LLM extraction for entities, relations, attributes, and optional summary nodes. *Load* writes to a triple store, a relational store, and a vector index, supporting hybrid traversal and semantic retrieval.

A.2.4 T3 — Adaptive / Self-Reflective Retrieval

Self-RAG (Asai et al., 2024). Self-RAG fine-tunes a single language model to emit four *reflection tokens* interleaved with normal text, indicating whether to retrieve, whether a passage is relevant, whether an answer is supported, and whether the passage is useful. Training data is bootstrapped from a critic model. At inference, the model retrieves on demand, generates per passage in parallel, self-grades each candidate via reflection-token probabilities, and selects the best continuation.

[1] Official Cognee repository: https://github.com/topoteretes/cognee.

A.2.5 T4 — STATEFUL AGENTIC MEMORY

T4 systems maintain a persistent, mutable memory store beside the LLM, with explicit write, update, decay, and read operations. They differ on three orthogonal axes: representation, which may be flat, graph-based, or verbatim; write policy, which may consolidate or preserve; and read policy, which may use one-shot lookup or multi-step deep research.

SimpleMem (Liu et al., 2026). A lifelong-memory framework whose design thesis is that dialogue history can be compressed by orders of magnitude *without* losing answer-relevant content. A three-stage pipeline extracts semantic units, compresses them under a lossless criterion, and stores the deduplicated representation.

MemPalace. MemPalace[2] preserves conversation traces as verbatim chunks for later retrieval. We evaluate two modes over the same ChromaDB store: **MemPalace-Raw** uses vector search, while **MemPalace-Hybrid** combines vector distances with keyword overlap at a fusion weight of 0.30. Both modes use the same embeddings, retrieve 20 candidate chunks, and pass the top 5 to the same LLM for answer generation. Neither mode uses an LLM reranker. This comparison isolates the contribution of keyword fusion under a fixed store and answer-generation setup.

REMem (Shu et al., 2026). REMem explicitly targets *episodic* recollection rather than semantic-only memory. Offline, raw experiences are converted into a hybrid graph that links time-aware "gist" nodes, or episode summaries with temporal and spatial markers, with fact triples extracted from the same episodes. Online, an LLM-based "agentic retriever" performs iterative, multi-step retrieval and reasoning over the graph, with explicit refusal behaviour for unanswerable questions.

GAM (Yan et al., 2025). General Agentic Memory rejects fully *static* pre-built memory in favour of *just-in-time* synthesis. A *Memorizer* runs offline, distilling history into a small set of "highlights" while preserving the complete trace in a universal page-store. A *Researcher* runs online: given a request, it iteratively retrieves, reads, and integrates pages from the page-store, guided by the highlights, until it has synthesized a task-specific context.

B PERDOCBENCH DATA CONSTRUCTION DETAILS

This appendix collects the mechanics, hyperparameters, count tables, and quality-control filters behind the example-driven construction overview in §4. It is organized stage by stage: persona seed and domain coverage in §B.1, per-persona statistics in §B.2, property-graph schema in §B.4, question generation in §B.5, anchor-conversation generation in §B.6, cross-source assembly in §B.7, and the cross-dataset quality assessment in §B.8. Figure 5 provides the basic flow diagram for these construction stages.

B.1 PERSONA SEED AND DOMAIN COVERAGE

PerDocBench is instantiated with 10 expert personas, each a full academic profile (free-text biography, research focus, tenure stage). The personas span the full breadth of academic specialties: from mechanical and industrial engineering through materials and thermal-radiation physics, cosmology, four flavours of computer science (urban informatics, storage systems, agricultural computer vision, data science), to chemistry and mathematical typesetting. Personas are not stereotypes: each is a multi-line biography with a clear research agenda that the construction pipeline turns into search intent for paper retrieval and dialogue grounding.

Domain tagging. We tag each persona's research interests against the Wikipedia *Outline of academic disciplines* taxonomy, producing both a top-level domain, such as *Applied science* or *Outline of physics*, and a fine-grained subdomain, such as *Engineering and technology* > *Materials science*. Across the 10 personas, the corpus covers 10 top-level domains and 41 distinct fine-grained subdomains. Figure 6 shows the top-level distribution. Applied science, with 13 tags, and Computer science, with 12, dominate, consistent with the predominance of STEM faculty personas, while the corpus

[2]Official MemPalace repository: https://github.com/mempalace/mempalace.

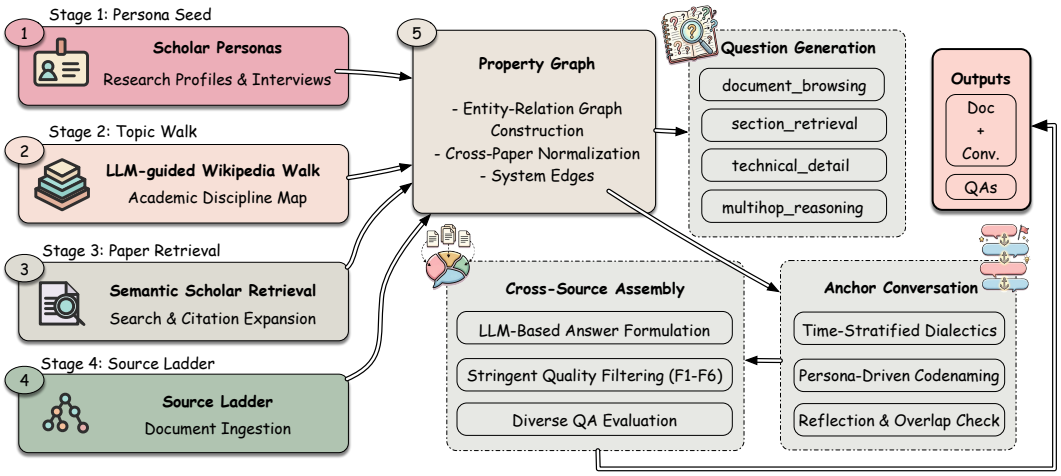


Figure 5: Basic PerDocBench data-construction pipeline. The flow starts from persona specification and paper retrieval, builds per-persona document and graph resources, generates document-grounded questions and anchor conversations, assembles cross-source instances, and applies quality filtering before release.

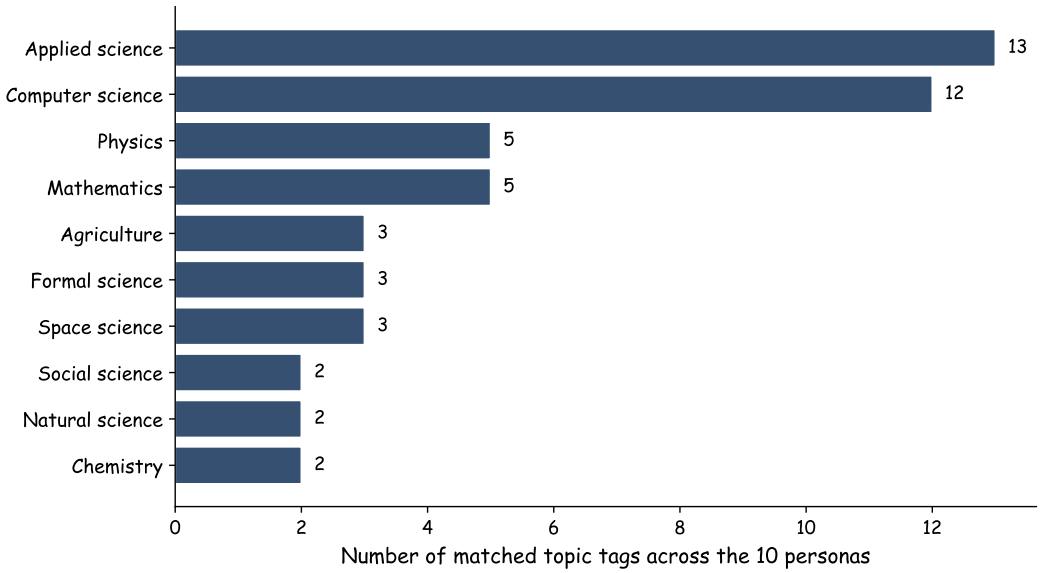


Figure 6: Top-level academic-domain coverage of the 10 PerDocBench personas. Bars count (persona, matched topic-tag) pairs from the topic matcher; ten top-level domains are represented, with a long tail into space science, chemistry, agriculture, the formal sciences, and the social sciences.

still contains visible footprints from physics, mathematics, space science, chemistry, agriculture, the formal sciences, and the social sciences. Bars count $\#\{(\text{persona}, \text{matched topic-tag})\}$ pairs. A persona contributes once per matched tag, so the bar total exceeds 10.

Topic-walk and retrieval settings. For each persona, OAI-5.4-mini runs the LLM-guided walk over the Wikipedia taxonomy with branching factor $B = 3$ and maximum depth $D_{\max} = 5$, producing a grounded topic shortlist rather than an unconstrained keyword expansion. We then issue $n_q = 5$ Semantic Scholar queries per persona, keep the direct seed hits, and add a one-hop citation expansion. The aggregate retrieval counts, year range, venue coverage, and seed-versus-expansion split are reported in §B.2 and Table 7.

Persona-to-domain table. Table 6 unpacks domain coverage further, listing each persona's primary top-level domain and its three highest-scoring fine-grained subdomains. No two personas share an identical subdomain triple, illustrating the breadth of expertise represented in the benchmark.

ID	Research persona (one-line summary)	Primary domain	Top fine-grained subdomains
P1	Computer-vision research scientist, agriculture	*Applied science*	Agricultural production / Agricultural engineering / AI
P2	Mechanical-engineering professor, post-grad mentor	*Applied science*	Mechanical engineering / Materials science / Systems science
P3	Data-science professor, digital content analysis	*Computer science*	Algorithms & data structures / Data & data structures / AI
P4	CS professor, urban informatics & smart infrastructure	*Computer science*	Concurrent / distributed systems / AI / Software engineering
P5	Chemistry professor, chemical product development	*Chemistry*	Chemical engineering / Materials science / Chemistry
P6	Thermal-radiation physics professor	*Physics*	Physics branches / Measurement / Statistics
P7	CS professor, storage technologies	*Computer science*	Databases / Data structures / Algorithms
P8	Cosmology professor emeritus	*Space science*	Cosmology / Astronomy / Space exploration
P9	Mathematics professor, typesetting workflow	*Mathematics*	Algebra / Calculus & analysis / Geometry & topology
P10	Industrial-engineering professor, simulation methods	*Applied science*	Materials science / Chemical engineering / Mechanical engineering

Table 6: The 10 research personas and their tagged scientific coverage. Each row shows the persona ID (also used in Table 7), a one-line summary, the dominant top-level domain (Wikipedia *Outline of academic disciplines*), and the three highest-scoring fine-grained subdomains returned by our topic matcher.

B.2 Per-persona Detailed Statistics

The benchmark construction pipeline builds, for each persona, (i) the persona biography, (ii) a per-persona library of real-paper metadata and accessible document renderings, and (iii) the conversation trace that wraps everything in a long personal-memory context. The release is intended to expose the derived benchmark instances, metadata, prompts, and rehydration scripts; full-text PDFs or parsed full-text renderings are redistributed only when their source access conditions permit it. A per-persona property-graph index is also built over those papers during construction, but it is an internal QA-design scaffold (used at dataset-construction time only, especially to seed multi-hop questions) and is *not* part of the released benchmark and *not* shown to the model under test.

Corpus counts. The corpus comprises 10 research personas indexing 677 real Semantic-Scholar papers. This corresponds to a mean of 67.7 papers per persona, with a range of 26–133. The corpus includes 629 direct topic-query hits and 48 one-hop citation expansions. Of these papers, 112 have openly accessible PDFs that we successfully download, and 97 are parsed and embedded into the per-persona property graph. The remaining 15 fall back to title+abstract rendering during graph construction. The 97 figure counts parsed PDFs, as in the *PDFs* column of Table 7; it does not count the full metadata library. Across the 10 shards, the benchmark releases 266 question–answer pairs, with a mean of 26.6 per persona and a range of 20–30. Per-shard memory-context lengths span 21k–72k whitespace tokens, with a mean of approximately 44.0k tokens, so every instance is firmly in the long-context regime while still fitting inside the OAI-4o-mini backbone window without truncation.

Question-type mix. The four reasoning families are distributed as follows: *Technical Detail* has 110 items, or 41.4%; *Section Retrieval* has 66 items, or 24.8%; *Multihop Reasoning* has 48 items, or 18.0%; and *Document Browsing* has 42 items, or 15.8%. All released items also carry a non-empty query rewrite, so the benchmark can double as a query-rewriting evaluation.

Per-persona breakdown. Table 7 reports raw construction counts. Its XS-Q column contains 292 assembled candidates before final release filtering; the released evaluation set contains 266 questions. The persona-to-discipline mapping is as follows: P1 is agricultural computer vision, P2 mechanical engineering, P3 data science, P4 urban informatics, P5 chemical engineering, P6 thermal-radiation physics, P7 storage and databases, P8 cosmology, P9 pure mathematics, and P10 industrial engineering.

Table 7: Per-persona raw statistics for PerDocBench. *Papers*: Semantic-Scholar-retrieved papers in the persona's library; *Seed*: papers retrieved by direct query (the rest are 1-hop citation expansion); *PDFs*: parsed into the property graph; *DocQ*: document-QA pairs; *XS-Q*: assembled cross-source QA candidates before final release filtering; *Anchor*: anchor-conversation segments per persona; *Topics*: matched fine-grained topic tags.

	Papers	Seed	PDFs	DocQ	XS-Q	Anchor	Topics
P1	39	39	7	33	33	8	5
P2	105	105	17	33	30	8	5
P3	79	79	10	33	28	8	5
P4	133	133	12	33	31	8	5
P5	70	70	11	33	31	8	5
P6	70	70	8	33	32	8	5
P7	26	3	9	29	25	8	5
P8	27	2	5	29	22	8	5
P9	79	79	9	33	30	8	5
P10	49	49	9	33	30	8	5

B.3 Evidence-Density Ratio

PerDocBench pairs four research-focused conversation segments with a much longer document corpus. We use a lexical proxy to describe how concentrated answer-related terms are in these two sources. This statistic characterizes the benchmark; it is not an evaluation metric or a claim that the conversation contains the answer.

Definition. For question q, let y_q be the gold answer and let $p(q)$ denote its persona. We write $C_{p(q)}$ for the concatenation of that persona's four research-conversation segments and $K_{p(q)}$ for the corresponding document corpus. Let $g(x)$ return the set of content-word types obtained by lowercasing x, retaining word forms that begin with a letter and have at least three characters (with digits, hyphens, and apostrophes allowed thereafter), and removing a fixed English stop-word list. Let $N(S)$ be the number of whitespace-delimited tokens in source S. We define the *answer-term density*

$$d(q, S) = \frac{|g(y_q) \cap g(S)|}{N(S)}, \qquad S \in \{C_{p(q)}, K_{p(q)}\}.$$

The numerator counts distinct gold-answer terms that occur at least once in a source; repeated occurrences do not increase the score. Our calculation script contains the exact normalization rule and stop-word list.

We compute $d(q, S)$ separately for every question in the full set Q of 266 questions. EDR divides the mean research-anchor density by the mean document-corpus density:

$$\mathrm{EDR} = \frac{\frac{1}{|Q|}\sum_{q \in Q} d\big(q, C_{p(q)}\big)}{\frac{1}{|Q|}\sum_{q \in Q} d\big(q, K_{p(q)}\big)}.$$

Source lengths in Table 8 are averaged separately over the ten personas.

Interpretation. The answer terms are $6.5\times$ more concentrated in the four research-anchor segments than in the document corpus (Table 8). The corpus nevertheless contains more of the gold answer in absolute terms: on average, 7.8 of 8.3 distinct answer terms occur in the corpus, compared with 3.1 in the research segments. The difference comes from source length—the corpus is roughly $21\times$ as long. EDR therefore measures lexical concentration, not evidence completeness; the document corpus remains the source of the gold answer.

Table 8: Lexical concentration in the research-anchor segments and document corpus. EDR compares their mean answer-term densities over all 266 questions. It measures concentration, not which source contains the complete supporting evidence.

Quantity	Value
KB corpus length (mean per persona)	40.8k
Research-anchor length (mean per persona)	≈2.0k
Corpus / anchor length ratio	20.9×
Mean $d(q, K_{p(q)})$ over Q	0.023%
Mean $d(q, C_{p(q)})$ over Q	0.153%
EDR (anchor mean / corpus mean)	6.5×

B.4 PROPERTY-GRAPH SCHEMA

Source selection and chunking. For each paper p, we pick the highest-fidelity available rendering. Parsed Markdown is used when available, otherwise the PDF is used, and title-plus-abstract is used only as a fallback. The chosen source is sentence-split into chunks $\{c_{p,j}\}$ of approximately CHUNK-SIZE tokens with OVERLAP tokens of overlap. Every chunk's metadata records its parent paper identifier, year, and source tier.

Schema-guided extraction. Each chunk $c_{p,j}$ is passed to a schema-LLM path extractor that returns typed triplets $(s, \tau_s) \xrightarrow{r} (o, \tau_o)$ with $\tau_s, \tau_o \in \mathcal{V}_{\text{ent}}$ and $r \in \mathcal{V}_{\text{rel}}$, restricted to 23 explicitly enumerated (τ_s, r, τ_o) patterns. The extractor is run in strict mode, so off-vocabulary triplets are rejected at decode time, and a post-hoc filter sweeps the persisted graph one more time.

Cross-paper structure. Entity labels are normalized: when paper p_1 says METHOD: contrastive learning and paper p_2 says the same, the two chunks attach to the same METHODOLOGY node. Combined with system edges SOURCE, NEXT, PREVIOUS that tie each chunk to its parent paper and to its narrative neighbours, multi-paper paths

$$c_{p_1,j} \xrightarrow{\text{SOURCE}} \underbrace{p_1 \xrightarrow{r_1} e \xrightarrow{r_2} p_2}_{\text{cross-paper}} \xrightarrow{\text{SOURCE}^{-1}} c_{p_2,k} \tag{1}$$

become first-class citizens of the corpus and are exploited by the multi-hop branch of question generation.

Table 9: Property-graph schema. Hyper-parameters: extractor LLM OAI-4o-mini at temperature 0; embeddings OAI-Emb-3-Small; chunk size 2048, overlap 200; SchemaLLMPathExtractor in strict mode.

Vocabulary	Members
Entities (10)	RESEARCH_TOPIC, METHODOLOGY, DATASET, HYPOTHESIS, FINDING, METRIC, LIMITATION, FUTURE_WORK, RELATED_WORK, CONTRIBUTION
Relations (12)	USES_METHOD, EVALUATES_ON, ADDRESSES_TOPIC, PROPOSES_HYPOTHESIS, REPORTS_FINDING, MEASURES_WITH, ACKNOWLEDGES_LIMIT, SUGGESTS_FUTURE, CITES, IMPROVES_UPON, CONTRADICTS, CONFIRMS
Valid patterns (23)	METHODOLOGY $\xrightarrow{\text{EVALUATES_ON}}$ DATASET; METHODOLOGY $\xrightarrow{\text{MEASURES_WITH}}$ METRIC; FINDING $\xrightarrow{\text{CONFIRMS}}$ HYPOTHESIS; FINDING $\xrightarrow{\text{CONTRADICTS}}$ HYPOTHESIS; METHODOLOGY $\xrightarrow{\text{IMPROVES_UPON}}$ METHODOLOGY; RELATED_WORK $\xrightarrow{\text{CITES}}$ METHODOLOGY; METHODOLOGY $\xrightarrow{\text{ACKNOWLEDGES_LIMIT}}$ LIMITATION; CONTRIBUTION $\xrightarrow{\text{ADDRESSES_TOPIC}}$ RESEARCH_TOPIC; (15 more in code release)

B.5 QUESTION-GENERATION DETAILS

We generate four question types, each with its own filtering pipeline.

Document Browsing. Per-paper LLM topic extraction is run once and cached. Topics that cover at least two papers are kept, and a question of the form "*what papers do I have about t?*" is generated. The benchmark default emits five browsing questions per persona.

Section Retrieval. A target section is inferred from chunk metadata or by regex over section names such as methods, results, discussion, and abstract. We enforce a $75\%/25\%$ non-abstract to abstract distribution. An LLM rejects generations whose ground truth degenerates to "not provided." The default is eight Section Retrieval questions per persona.

Technical Detail. We sample one filtered triplet (s, r, o) and require o to be a *proper or rare noun*. A blocklist removes generic terms such as "accuracy," "method," and "system." We then ask a single-hop fact question whose answer is o. The default is 12 per persona.

Multihop Reasoning. We build 2-hop paths $(A, r_1, B), (B, r_2, C)$, reject cyclic and CITES-only paths, force the terminal C to be a proper noun, and bind the question word to r_2. For example, MEASURES_WITH maps to a "what metric" question. We require that at least 33% of the multi-hop instances have A and C originating in different papers, so that Multihop Reasoning genuinely crosses documents. The default is eight per persona, with up to 50 retries.

Question-form constraints. Across all four types, the generated questions must be self-contained: no paper title may appear, and phrases such as "the paper," "the study," or "this work" are disallowed. Question length is 25–35 words, and generation temperature is 0.0. The four-question-type design itself is part of the attack on the unified-context form. A system that overfits to one question pattern, such as short-fact lookup, may fail on one or more of the other three: browsing, Section Retrieval, or Multihop Reasoning.

B.6 ANCHOR-CONVERSATION GENERATION

Anchor filtering. Anchor selection uses an implementation heuristic, not a scoring rule from prior work. We keep chunks that are long enough to contain usable context, can be traced back to a document identifier, and look like ordinary prose. We downweight chunks that resemble references, acknowledgments, author lists, figure/table debris, OCR noise, or equation-heavy blocks. The surviving chunks then pass a small LLM "assess-and-clean" step, which removes noisy text and emits a project codename plus 3–6 topical keywords.

Time-stratified generation. For each persona we generate four conversation segments anchored to distinct time frames: initial discussion, next week, next month, and next year. Each segment is a 4–6-turn dialogue that mentions the project codename, is subtly connected to the anchor excerpt without quoting or naming the paper, and includes at least two keywords verbatim. A post-processing pass merges consecutive same-speaker turns, deduplicates repeated blocks, and inserts a side-note timestamp after the first exchange so segments are temporally identifiable. A reflection LLM pass enforces narrative continuity by removing greetings between segments and maintaining the same project across the segment.

Codenames vs. paper titles. Codenames tie the conversation to the document KB without ever *naming* the paper: when the conversation later mentions "Project Magnolia," resolving the reference entails identifying the time period in which the name was introduced and binding it to a specific paper in the KB. By design, the conversation segments are informationally insufficient: without consulting the document KB, the agent cannot recover paper-level facts. This is the lever pulled by the cross-source family.

B.7 Cross-Source Assembly and the F1–F6 Quality Filter

A cross-source item is built by combining a conversation stem with a document question and asking an LLM to rewrite the joined prompt with a transition connector such as "given that," "in that context," or "from that project context." The stem of the rewritten question lives in the conversation through the project codename, time period, and topical cue. The answer lives in the document KB. Neither half is sufficient on its own.

Anchor matching per question. For each document question, we choose a conversation anchor by a simple matching rule. We first prefer anchors from the same gold paper as the source question. If several anchors remain, we rank them by lexical overlap with the source question, its gold answer, and its browsing-topic filter. The top two anchors are tried in order. The conversation context is then sliced using side-note timestamps, so the rewritten question is paired with the time period in which that project was discussed.

Six-detector quality filter. The LLM rewrite is run through six failure detectors:

F1 The question must mention the target section for SECTION_RETRIEVAL items.

F2 The question and at least one of evidence / answer / rewrite must overlap anchor tokens.

F3 At most 45% of content tokens may be unseen by the union of anchor and source question.

F4 Inventory phrasing is required for BROWSING and forbidden elsewhere.

F5 The rewrite must overlap the original benchmark question or answer.

F6 The ground truth must not appear verbatim in the question.

For 70% of items we additionally require the project codename or at least two event tokens to explicitly appear in the question. Each item is given up to 3×2 attempts, corresponding to three rewrites over two candidate anchors. Items with more than one residual failure are dropped.

B.8 Cross-dataset Quality Assessment Details

Setup. We compare PerDocBench with two commonly used personal-memory benchmarks, LoCoMo (Maharana et al., 2024) and LongMemEval-S* (Wu et al., 2024; Hu et al., 2025). The same assessment protocol is applied to all three datasets and reports two families of indicators: *deterministic structural quality*, computed directly from the question, answer, and supplied context, and *LLM semantic quality*, judged by DeepSeek-v4-flash on up to 300 sampled examples per dataset under a strict 0–2 rubric.

Deterministic family. *Grounded* (↑): rate of items whose gold answer appears in the context or has high answer–context token overlap. *Dup.* (↓): near-duplicate question rate from token-level Jaccard overlap within the same dataset split. *Context-dep.* (↑): rate of items that are grounded *and* have sufficient question–context or answer–context overlap. The aggregate **Overall Quality** is the unweighted average $Q = \frac{100}{3}\big(r_{\text{ground}} + (1 - r_{\text{dup}}) + r_{\text{ctx}}\big)$, with the duplicate rate flipped so all components share the higher-is-better convention.

LLM semantic family. The judge receives the question, gold answer, and up to six retrieved evidence snippets, and returns JSON scores on four axes. Each axis is a 0–2 integer: semantic evidence support, answerability, context necessity, and shortcut resistance. The **strict pass** criterion is derived deterministically from these axes. Both critical axes, semantic evidence support and answerability, must be at least 1, and the total score must be at least $6/8$. Issues are returned as short tags for unsupported answers, ambiguous questions, weak context need, and shortcut risk.

Single prompt for four axes. The four LLM axes are scored in a single judging pass rather than via separate prompts so the judge sees the same evidence and applies a consistent semantic standard across axes; this avoids the inconsistency that would arise if, e.g., *Answerability* and *Context Necessity* were evaluated against different snippet selections. The strict pass is then derived deterministically and does not require an additional LLM call.

Findings. Table 10 reports every individual axis, not only the aggregate score. PerDocBench leads on every deterministic metric (highest grounded rate 98.12%, lowest near-duplicate rate 0.75%, highest context-dependence rate 98.12%, highest Overall Quality 98.50) and on every LLM semantic axis (evidence support 1.119 vs. 0.221 and 0.045; answerability 1.303 vs. 0.384 and 0.239; context necessity 1.373 vs. 0.834 and 0.541; shortcut resistance 1.267 vs. 0.835 and 0.667), and the highest strict pass rate (35.00% vs. 6.76% and 1.35%). The assessment supports a quality-over-quantity interpretation: although PerDocBench is smaller than LoCoMo in raw item count, it leads on every quality axis we measure, suggesting its examples are more likely to require the intended memory context rather than be answerable through superficial cues.

Table 10: Cross-dataset quality assessment. Arrows denote the preferred direction; **bold** marks the best per row. PerDocBench leads on the aggregate score and on every individual LLM semantic axis.

Metric	PerDoc	LoCoMo	LME-S*
Deterministic structural metrics			
Grounded (↑)	**98.12%**	88.71%	80.00%
Duplicate rate (↓)	**0.75%**	2.47%	1.33%
Context-dep. (↑)	**98.12%**	88.71%	80.00%
Overall Quality (↑)	**98.50**	91.65	86.22
LLM semantic metrics (judge: DeepSeek-v4-flash)			
Evidence support /2 (↑)	**1.119**	0.221	0.045
Answerability /2 (↑)	**1.303**	0.384	0.239
Ctx. necessity /2 (↑)	**1.373**	0.834	0.541
Shortcut resistance /2 (↑)	**1.267**	0.835	0.667
Strict pass (↑)	**35.00%**	6.76%	1.35%

B.9 Distractor-rewrite attack

For completeness, Table 11 reports the same numbers shown in Table 2 (§3) in tabular form.

Table 11: Density-rewriting attack. Flip ratio $\rho_m = N_{R\to W}(m)/N_R(m)$ on LongMemEval-S* (LME-S*) and LoCoMo for two methods and two metrics. Higher means a larger fraction of originally-correct answers are broken by an answer-preserving density rewrite.

Method	Metric	LME-S*	LoCoMo
HippoRAG-v2	Exact match	13.8%	35.5%
HippoRAG-v2	ROUGE-L F1	15.0%	26.5%
MemPalace	Exact match	28.6%	**63.6%**
MemPalace	ROUGE-L F1	36.4%	**50.8%**

B.10 Artifact Sources, Licenses, and Intended Use

Paper-source acquisition. For each scholar persona, we derive topic queries from the matched academic disciplines and retrieve real papers through a Semantic Scholar-compatible literature API. We retain papers with abstracts and store standard bibliographic metadata, source links, citation links, and open-access PDF pointers when provided. For construction only, each paper is rendered from the best available local source: parsed text when available, otherwise an accessible PDF, and otherwise title-plus-abstract text.

Access terms and redistribution. We use Semantic Scholar metadata and open-access pointers under their applicable API and source-document terms, and we cite the benchmark, model, retrieval, and memory-system artifacts used in the experiments. An open-access PDF link is treated as evidence that the paper can be accessed for research processing, not as a general license to redistribute the full text. Accordingly, the intended PerDocBench release consists of derived benchmark instances, metadata, prompts, source links, and construction code; full-text papers or parsed full-text renderings are included only when redistribution is permitted by the original source.

Intended use. PerDocBench is intended for non-commercial research evaluation of personalized memory, retrieval, and long-context systems. It is not intended for resale of third-party scholarly data, redistribution of restricted source documents, construction of deployable user profiles, or evaluation on private user data without consent. The scholar personas and anchor conversations are benchmark contexts, not real user records.

Identifier and content checks. The source papers are public scholarly works and therefore contain author names and other bibliographic identifiers; we keep these only as citation metadata, not as private user attributes. The generated personas, conversations, questions, and answers are screened for unsupported content, answer leakage, unrelated personal identifiers, and offensive or unsafe generations. Flagged items are removed or rewritten before release.

C Supplemental Experiments and Results

This appendix collects the full protocol for the distractor-rewrite hackability study of §3 (§C.1), the per-method adaptation notes (§C.2), the metric definitions used by all result tables (§C.3), a validation study of the LLM judge against human annotators and an alternative judge model (§C.4), the full $17 \times$ all-metrics Main and Ablation tables (§C.5), a long-context backbone sensitivity comparison (§C.7), the extended case study (§C.8), and an analysis of answer-string matches and correctness for methods with alignable retrieval artifacts (§C.9).

C.1 Distractor-Rewrite Hackability Study

This subsection expands on the motivation experiment in §3, providing the full rewriter protocol, sample-selection rules, formal flip-ratio definitions, and the rationale for our method and metric choices. Summary numbers are listed in Table 2.

Rewriter and answer-preservation check. For each evaluated example, we prompt a separate LLM, independent of the memory method under test, to rewrite the supporting-evidence utterance with lower information density. The prompt asks it to rephrase the utterance while preserving support for the answer, leave the question and gold answer untouched, and insert 2–3 plausible but irrelevant side details. These details may involve a different time, a place, a side errand, a coffee break, or a brief mood note. The rewriter also flags whether the rewritten evidence still supports the answer; outputs that fail this self-check are discarded. Retained variants thus pass an answer-preservation check, but this check is not independent verification and can miss changes in support. We interpret the experiment as a presentation-sensitivity probe under that limitation. The full rewriter prompt is in Appendix D.4.

Sample size and selection. We apply the rewrite to 69 of 300 LongMemEval-S* questions and 300 of 1,540 LoCoMo questions ($\approx 19\%$–23% of each benchmark). The slice is chosen large enough to materially perturb the utterance-density structure across each benchmark—the rewrites cover the dominant question-type distribution of both benchmarks—yet small enough that each benchmark remains readable as a whole and the rewriter API cost stays bounded.

Method choice. We stress-test two memory designs: **HippoRAG-v2** (T2, structure-induced retrieval over a constructed graph) and **MemPalace-Raw** (T4, retrieval from a persistent memory store). Observing flips in both shows that presentation sensitivity in the tested benchmark slices is not confined to one implementation. It does not establish how all methods would respond to the same intervention.

Flip ratio: formal definition. Let $(\hat{y}_i^{\text{base}}, \hat{y}_i^{\text{var}})$ denote the paired predictions on example i with gold answer y_i, and let $m(\cdot,\cdot) \in [0,1]$ be one of the two metrics. We binarize each criterion at a fixed threshold $\tau = 0.5$ (a prediction is *correct* if its score $\geq \tau$, otherwise *wrong*; this is trivially the native form for EM but is required for ROUGE-L F1 since it is continuous in $[0,1]$), defining a single per-example correctness indicator parameterized by the run s:

$$c_i^s(m) \;=\; \mathbb{1}[\, m(\hat{y}_i^s, y_i) \geq \tau \,], \qquad s \in \{\text{base}, \text{var}\}.$$

From these we derive the cell-level counts—the number of baseline-correct examples, and the number that move from **R**ight (correct on the original) to **W**rong (wrong on the variant)—and the flip ratio:

$$N_R(m) = \sum_i c_i^{\text{base}}(m),$$
$$N_{R\to W}(m) = \sum_i c_i^{\text{base}}(m)\left(1 - c_i^{\text{var}}(m)\right),$$
$$\rho_m = \frac{N_{R\to W}(m)}{N_R(m)} \in [0,1].$$

ρ_m, the hack-sensitivity flip ratio for metric m, is the fraction of originally correct cases that become incorrect after a retained rewrite. It measures sensitivity to this intervention under the answer-preservation check above. We report ρ_m rather than the symmetric agreement $\Pr[c^{\text{base}} = c^{\text{var}}]$ to focus on the loss of originally correct predictions.

What a high ρ_m does and does not imply. A high ρ_m indicates sensitivity to evidence *packaging* in the tested method–benchmark pairs. The intervention changes length and wording along with density, so it does not isolate a single mechanism or imply a general inability to reason robustly. PerDocBench (§4) assigns complementary roles to conversation and document evidence, so answering requires cross-source binding rather than exploiting one compact evidence format.

C.2 Per-Method Adaptation Notes

After the hackability protocol above, we record the benchmark-specific adaptations, model choices, and retrieval knobs used by the two-stage MemoryAgentBench interface (Hu et al., 2025); the mechanisms of the baseline families are summarized in §A.2. Unless otherwise noted, methods read the benchmark's 4096-token chunks, return evidence to the shared OAI-4o-mini answer reader, and keep all method-specific state under a per-context runtime directory rather than an upstream global path.

Run protocol and descriptive statistics. The main corpus-first condition is evaluated over **three independent runs** per system. All accuracy columns in Table 3 report mean ± sample standard deviation (n=3). The answer-string matching analysis uses the corpus-first run for which intermediate artifacts were saved. This is one of the three summarized runs, but it is not the run shown in Table 13, so their per-method correct rates need not match. The anchor-first ablation, lexical metrics, and efficiency columns are each reported from one completed run. We treat the ordering ablation as a single-run paired contrast and do not attach cross-run uncertainty. Wall-clock timings are also single-run measurements because they vary with machine load independently of answer correctness. Correctness entries are corpus-level means over all examples in a run, not maxima over trials.

OAI-4o-mini (T0). The memory-write step is a no-op. At query time, the full persona evidence stream is concatenated into the prompt, with generation length 256 and temperature 0.

BM25 (T1). We use the same 4096-token chunking as the dense retrievers and retrieve the top 10 chunks. This keeps the sparse baseline aligned with the T1 dense baselines except for the scoring function.

Contriever (T1). The shared FAISS wrapper loads the local `facebook/contriever` checkpoint. Document and query vectors use the encoder's first-token representation with L_2 normalization, and retrieval depth is top 10.

OAI-Emb-3-Small / OAI-Emb-3-Large (T1). We use the OAI API `OAI-Emb-3-Small` and `OAI-Emb-3-Large` models with their native 1,536- and 3,072-dimensional outputs. Both use the same FAISS wrapper and top-10 retrieval depth, so the pair isolates the embedding-backbone change.

Qwen3-Embedding-4B (T1). The same FAISS wrapper loads `Qwen/Qwen3-Embedding-4B`. It mean-pools the final hidden states under the attention mask, applies L_2 normalization, and retrieves the top 10 chunks.

RAPTOR (T2). We keep the reference-style tree depth at three levels and retrieve top-10 evidence for the answer reader. The reported construction time is dominated by offline tree construction, which accounts for the 264.98 s memory-construction time.

GraphRAG (T2). The wrapper runs GraphRAG with default Leiden/community-report settings and passes top-10 retrieved evidence to the shared answer reader. Its 28.58 s per-query time is the highest in the suite, mainly because query answering invokes the framework's hierarchy selection and map-reduce path.

HippoRAG-v2 (T2). Our main run uses the OAI-Emb-3-Small variant. The wrapper feeds metadata-free benchmark chunks into the upstream graph/retrieval stack and returns the top 10 chunks to OAI-4o-mini.

MemoRAG (T2). We configure MemoRAG with a Qwen2-7B memory checkpoint (`TommyChien/memorag-qwen2-7b-inst`) and BGE-M3 retriever (`BAAI/bge-m3`). The retrieval hit count is 3 and the retrieval chunk size is 4096; saved MemoRAG memory is reused by context id when a cached `memory.bin` is present.

Cognee (T2). We run the text-only ECL ingestion path and leave absent metadata fields empty. The chat-completion-compatible OAI-4o-mini and OAI-Emb-3-Small settings are mirrored into Cognee's environment variables, and the query returns top-10 evidence.

Self-RAG (T3). Each chunk is wrapped as a LangChain document with only synthetic `source` and chunk-index metadata. We set `top_k=10`; the high construction cost (352.29 s, the highest in our suite) comes from per-passage generation and self-grading.

MemPalace-Hybrid (T4). We evaluate the generalizable `hybrid_v1` path: ChromaDB's default all-MiniLM-L6-v2 embeddings, retrieval depth 20, answer-context depth 5, and keyword-overlap fusion weight 0.30. Drawer text is split at roughly 4,000 characters so that benchmark chunks remain embeddable.

MemPalace-Raw (T4). Raw uses the same ChromaDB store, drawer split, embedding function, and answer-context depth as MemPalace-Hybrid, but disables the keyword-overlap fusion. We omit the upstream palace, room, closet, and KG variants because they require person names, session dates, mined project files, or structured triples that the two-stage benchmark interface does not expose.

REMem (T4). The wrapper ports the episodic-gist path only. It buffers raw chunks during memory writing, extracts verbatim/gist/fact elements on first query, replaces the upstream NV-Embed-v2 dependency with OAI-Emb-3-Small, and uses OAI-4o-mini for both extraction and QA. The upstream graph walk and DSPy reranker are not used; dense retrieval keeps top-20 elements and feeds the top 5 to the answer prompt.

SimpleMem (T4). Each benchmark chunk is treated as one user dialogue item with no timestamp, and automatic timestamp injection is disabled. We use the API backend with OAI-4o-mini, `Qwen/Qwen3-Embedding-0.6B` embeddings, window size 4, overlap size 1, semantic/keyword/structured top-k values of 20/5/5, and at most two reflection rounds. Streaming is disabled, and LanceDB storage is rooted in the per-context runtime directory.

GAM (T4). We use the upstream text workflow only. The default mode buffers all benchmark chunks and performs a single batched `add` on the first query, which avoids repeatedly rebuilding the taxonomy across chunks. Docker/video dependencies are soft-imported away, the memory builder uses four workers, and the request agent is capped at 20 exploration iterations.

C.3 Metric Definitions

All metrics compare a method output $\hat{y}_i$ against the gold answer set Y_i for example i after the common answer-extraction step. When multiple gold strings are available, lexical metrics take the maximum

score over $y \in Y_i$. We report corpus-level means, multiplying all correctness and overlap scores by 100 in tables; the only unscaled columns are the runtime columns measured in seconds. The primary semantic judge is OAI-4o, used with a fixed rubric, temperature 0, and a JSON-constrained output schema.

L1 and L2 judge correctness. The two primary columns, **L1** and **L2**, are binary LLM-as-a-judge correctness rates. For each example, OAI-4o receives the original question, the question type, the gold answer, retrieval/tool metadata when relevant, and the model response. It returns two nested indicators, $\ell_i^{(1)}, \ell_i^{(2)} \in \{0, 1\}$. The lenient indicator $\ell_i^{(1)}$ is 1 when the response contains the requested gold fact or an unambiguous equivalent. The strict indicator $\ell_i^{(2)}$ is 1 only when the response is also precise, task-faithful, and free of unsupported or conflicting content. Thus L1 is lenient semantic correctness: it gives credit for recovering the essential answer even when wording differs or minor presentation details vary. L2 is the strict criterion: it requires the same core correctness while penalizing hallucinated documents, unsupported extra details, contradictions, missing gold items in list-style answers, and generic responses that do not resolve the requested memory fact. We enforce $\ell_i^{(2)} \leq \ell_i^{(1)}$ during parsing, so strict correctness is a subset of lenient correctness. The reported scores are

$$\mathrm{L1} = 100 \cdot \frac{1}{N} \sum_{i=1}^{N} \ell_i^{(1)}, \qquad \mathrm{L2} = 100 \cdot \frac{1}{N} \sum_{i=1}^{N} \ell_i^{(2)}.$$

These judge-based metrics are the primary metrics because PerDocBench answers often require semantic equivalence, title-level coverage, or multi-step integration that surface-form metrics cannot reliably distinguish.

Lexical overlap metrics: EM, F1, and RGL F1. The secondary lexical metrics measure surface agreement after normalizing answers by lowercasing, removing punctuation and English articles, and collapsing whitespace. **EM** is exact match under this normalization:

$$\mathrm{EM}_i = \max_{y \in Y_i} \mathbb{1}[\mathrm{norm}(\hat{y}_i) = \mathrm{norm}(y)].$$

F1 is token-level overlap F1. For a prediction–gold pair, let P_i and G_i be the multisets of normalized prediction and gold tokens, and let $C_i = P_i \cap G_i$ be their multiset intersection. Precision and recall are

$$\mathrm{Prec}_i = \frac{|C_i|}{|P_i|}, \qquad \mathrm{Rec}_i = \frac{|C_i|}{|G_i|},$$

and $\mathrm{F1}_i = 2\mathrm{Prec}_i\mathrm{Rec}_i/(\mathrm{Prec}_i + \mathrm{Rec}_i)$ when the denominator is non-zero, otherwise 0. **RGL F1** denotes ROUGE-L F1: the F-measure induced by the longest common subsequence between $\hat{y}_i$ and y, computed with stemming and again maximized over gold variants. These lexical scores are useful for continuity with prior long-memory benchmarks, but they are intentionally secondary: a response can be semantically correct with low EM, or lexically similar while still adding unsupported content that L2 rejects.

Runtime metrics: Mem(s) and Q(s). **Mem(s)** is the mean wall-clock time spent in the memory-construction stage for a method on a persona context. It includes indexing, embedding, graph construction, summarization, consolidation, or other write-time operations required before the query is answered. Methods with no explicit memory-write module, such as the T0 long-context backbone or pure query-time retrieval setups, can therefore have near-zero memory-construction time. **Q(s)** is the mean wall-clock time for the query stage, including retrieval, reranking, agentic search, and the final answer generation. We keep these columns in seconds rather than percentages because they quantify efficiency, not correctness.

Per-question-type metrics: TD, SR, DB, and MR. The suffixes **TD**, **SR**, **DB**, and **MR** denote the four PerDocBench question types: Technical Detail, Section Retrieval, Document Browsing, and Multihop Reasoning. For each type q, we compute the same L1 and L2 judge rates on the subset

$I_q = \{i : q_i = q\}$:

$$\mathrm{L1}(q) = 100 \cdot \frac{1}{|I_q|} \sum_{i \in I_q} \ell_i^{(1)},$$

$$\mathrm{L2}(q) = 100 \cdot \frac{1}{|I_q|} \sum_{i \in I_q} \ell_i^{(2)}.$$

The table columns therefore expand as **TD-L1/TD-L2** for precise technical facts, **SR-L1/SR-L2** for section-targeted retrieval, **DB-L1/DB-L2** for document or paper browsing, and **MR-L1/MR-L2** for questions requiring evidence integration across multiple memory clues. These disaggregated metrics are not separate scoring rules; they are stratified versions of the same OAI-4o judge criteria, designed to reveal which memory operation each method fails under.

C.4 Judge Validation: Human and Cross-Model Agreement

Because all headline metrics are LLM-judged, we validate the OAI-4o judge on a stratified 150-item subset of the main corpus-first condition. The subset is stratified across the four question types and across three judge-verdict classes (accept, boundary cases with L1=1/L2=0, reject), deliberately oversampling the boundary class where the judge is most contestable, and spans ten methods from the backbone and strong flat retrievers to structured, stateful, and tail-failure systems. Two annotators, both members of the author team, independently re-labeled L1/L2 *blind* to the machine verdicts, using the same rubric text as the judge prompt. No external annotators or crowd workers were recruited for this validation. As a cross-model check, DeepSeek-v4-flash re-judged the same subset under the identical prompt, rubric, and parsing logic.

Table 12: Judge validation on a stratified 150-item subset. Agreement (and Cohen's κ) with the OAI-4o judge used throughout the paper; the pooled row adds a bootstrap 95% CI. Boundary cases (L1=1, L2=0) are oversampled, so these figures describe the selected subset rather than estimating full-population agreement.

Comparison	n	**L1 agreement (κ)**	**L2 agreement (κ)**
Annotator A vs. OAI-4o judge	150	94.7% (0.87)	86.7% (0.71)
Annotator B vs. OAI-4o judge	150	88.7% (0.72)	76.0% (0.45)
Pooled humans vs. OAI-4o judge	300	91.7% [88.3, 94.7]	81.3% [77.0, 85.7]
Annotator A vs. B (IAA)	150	84.7% (0.64)	77.3% (0.42)
OAI-4o vs. DeepSeek-v4-flash judge	150	91.3% (0.78)	88.0% (0.76)

Table 12 shows pooled human–judge agreement of 91.7% on L1 and 81.3% on L2. L2 judgments vary both between humans and the judge (κ=0.45–0.71) and between the two humans (κ=0.42), indicating that the strict criterion is less consistently applied on this subset. The alternative judge agrees with OAI-4o on 91.3%/88.0% of L1/L2 labels (κ=0.78/0.76). These checks support the consistency of many item-level judgments, but do not rule out shared biases or establish full-benchmark ranking stability under another judge.

C.5 Full Main and Ablation Tables

Tables 13 and 14 report the complete 17× (7 main metrics + 8 per-question-type metrics) single-run values for the corpus-first (Main) and anchor-first (Ablation) conditions respectively (the corpus-first run shown here is run 1 of the three main-condition runs summarized in Table 3). Methods are sorted by capability tier and within tier by L2 strict (descending). Column legend: **L1** = lenient judge correctness ×100 (%); **L2** = strict judge correctness ×100 (%); **EM** = exact match ×100 (%); **F1** = token-level F1 ×100 (%); **RGL_F1** = ROUGE-L F1 ×100 (%); **Mem(s)** = memory-construction time, seconds; **Q(s)** = per-query latency, seconds; **TD/SR/DB/MR** = per-question-type L1/L2 on Technical Detail / Section Retrieval / Document Browsing / Multihop Reasoning (×100, %). All correctness/F1 metrics throughout this paper are reported in %.

Table 13: Full Main results (corpus-first) on PerDocBench. All 17 systems (single run; three-run summary in Table 3). Scores are in %; memory-construction and query times are in seconds. The backbone row is bolded.

Method	Tier	L1	L2	EM	F1	RGL F1	Mem(s)	Q(s)	TD-L1	TD-L2	SR-L1	SR-L2	DB-L1	DB-L2	MR-L1	MR-L2
OAI-4o-mini (backbone)	T0	48.1	26.3	0.8	29.4	26.3	0.00	3.84	54.3	35.1	51.4	24.3	30.2	7.0	47.5	28.8
BM25	T1	47.4	28.6	1.1	31.0	28.3	0.00	4.69	56.4	40.4	52.9	31.4	20.9	2.3	45.8	25.4
OAI-Emb-3-Large	T1	50.0	28.2	1.5	30.9	28.1	1.12	3.95	57.4	38.3	61.4	27.1	18.6	2.3	47.5	32.2
OAI-Emb-3-Small	T1	47.4	27.8	0.4	30.0	26.8	1.02	3.97	58.5	39.4	52.9	27.1	20.9	2.3	42.4	28.8
Qwen3-Embed-4B	T1	49.2	26.3	1.5	30.8	28.4	0.04	3.89	53.2	37.2	58.6	28.6	25.6	0.0	49.2	25.4
Contriever	T1	30.1	17.3	0.4	21.6	19.9	0.01	3.80	44.7	30.9	24.3	11.4	14.0	2.3	25.4	13.6
Cognee	T2	51.9	33.8	0.0	30.2	27.3	83.87	12.82	59.6	43.6	51.4	28.6	41.9	27.9	47.5	28.8
HippoRAG-v2	T2	45.9	32.7	9.8	41.5	40.1	20.31	13.77	57.4	43.6	47.1	30.0	25.6	16.3	40.7	30.5
GraphRAG	T2	39.8	21.4	1.9	29.2	28.1	22.72	28.58	48.9	33.0	47.1	20.0	18.6	0.0	32.2	20.3
MemoRAG	T2	32.3	20.7	0.4	24.1	22.1	1.68	10.21	41.5	30.9	35.7	18.6	4.7	0.0	33.9	22.0
RAPTOR	T2	28.6	17.7	3.0	27.5	26.2	264.98	1.43	34.0	25.5	27.1	14.3	20.9	0.0	27.1	22.0
Self-RAG	T3	33.8	18.0	0.8	22.4	21.2	352.29	17.83	40.4	27.7	40.0	17.1	9.3	0.0	33.9	16.9
MemPalace-Hybrid	T4	47.4	29.7	1.1	28.4	26.0	0.51	3.89	56.4	42.6	54.3	28.6	14.0	0.0	49.2	32.2
MemPalace-Raw	T4	45.9	28.6	1.1	29.1	26.7	0.51	3.82	57.4	43.6	50.0	30.0	20.9	0.0	40.7	23.7
REMem	T4	41.7	27.1	0.0	27.4	25.3	0.09	9.54	51.1	37.2	38.6	20.0	27.9	14.0	40.7	28.8
SimpleMem	T4	32.7	22.2	7.1	36.3	35.5	30.67	21.73	34.0	30.9	28.6	15.7	37.2	2.3	32.2	30.5
GAM	T4	30.8	13.9	0.0	24.0	22.5	183.50	20.16	37.2	24.5	25.7	11.4	30.2	0.0	27.1	10.2

Table 14: Full Ablation results (anchor-first) on PerDocBench. Scores are in % and times in seconds, as in Table 13; this is the same 17 systems with the anchor-bearing conversation block at the start of the context.

Method	Tier	L1	L2	EM	F1	RGL F1	Mem(s)	Q(s)	TD-L1	TD-L2	SR-L1	SR-L2	DB-L1	DB-L2	MR-L1	MR-L2
OAI-4o-mini (backbone)	T0	60.2	33.1	1.9	32.8	30.3	0.00	3.93	61.8	44.5	74.2	31.8	45.2	4.8	50.0	33.3
BM25	T1	58.3	32.3	2.3	33.2	30.6	0.00	4.61	62.7	45.5	72.7	28.8	35.7	2.4	47.9	33.3
OAI-Emb-3-Large	T1	57.1	32.0	1.1	30.3	27.9	1.54	4.33	60.9	41.8	71.2	34.8	35.7	0.0	47.9	33.3
OAI-Emb-3-Small	T1	57.1	33.8	1.1	30.9	28.6	1.10	3.88	60.0	43.6	74.2	40.9	31.0	2.4	50.0	29.2
Qwen3-Embed-4B	T1	58.3	33.1	1.9	32.1	29.6	0.04	3.96	61.8	41.8	66.7	33.3	38.1	4.8	56.2	37.5
Contriever	T1	43.6	25.6	1.1	27.7	25.9	0.01	4.10	45.5	33.6	50.0	27.3	23.8	0.0	47.9	27.1
Cognee	T2	58.6	36.1	0.8	31.5	28.0	48.86	12.33	60.0	42.7	68.2	30.3	57.1	26.2	43.8	37.5
HippoRAG-v2	T2	51.1	35.0	13.9	43.3	41.9	10.14	13.83	51.8	41.8	60.6	36.4	42.9	19.0	43.8	31.2
GraphRAG	T2	43.2	23.3	1.9	29.0	28.2	12.55	27.71	45.5	30.9	50.0	24.2	31.0	0.0	39.6	25.0
MemoRAG	T2	51.1	30.5	1.5	31.0	28.9	1.06	9.19	52.7	38.2	47.0	30.3	50.0	0.0	54.2	39.6
RAPTOR	T2	30.5	19.2	6.0	28.5	27.4	230.36	1.42	39.1	26.4	27.3	19.7	19.0	0.0	25.0	18.8
Self-RAG	T3	36.8	23.3	1.9	24.3	23.4	382.61	19.02	41.8	30.0	42.4	24.2	14.3	0.0	37.5	27.1
MemPalace-Hybrid	T4	56.0	33.1	2.6	31.1	28.9	0.48	3.94	62.7	48.2	69.7	31.8	31.0	2.4	43.8	27.1
MemPalace-Raw	T4	53.0	32.3	2.6	30.5	28.1	0.49	4.01	61.8	48.2	62.1	28.8	28.6	2.4	41.7	27.1
REMem	T4	48.1	31.6	0.0	28.0	26.0	0.09	8.03	54.5	41.8	56.1	25.8	28.6	19.0	39.6	27.1
SimpleMem	T4	32.0	19.9	7.1	34.3	32.6	20.50	20.96	33.6	26.4	25.8	13.6	52.4	16.7	18.8	16.7
GAM	T4	29.7	12.4	0.0	21.2	20.2	162.24	19.82	34.5	17.3	28.8	15.2	23.8	0.0	25.0	8.3

C.6 Full Per-Metric Delta Table

Table 15 reports the per-method delta (anchor-first minus corpus-first) on every metric for the single-run anchor-position ablation. All values are shown in % and computed from the unrounded scores. Green cells indicate that the method improves when the anchor-bearing conversation block is placed at the start of the context; red cells indicate degradation.

C.7 Long-Context Backbone Sensitivity

We additionally evaluate five cloud long-context LLMs in the same T0 setting: the persona's full evidence stream is concatenated into a single prompt, with no retrieval or memory module, under both corpus-first (Main) and anchor-first (Ablation) orders. Data, prompts, judge model, decoding parameters, and seed are identical to the main runs; memory-construction time is therefore 0.0 s for all models.

Table 16 shows two patterns. First, newer or more expensive backbones are not automatically better. DeepSeek-v4-flash is the strongest overall and cheaper than OAI-4o-mini, whereas OAI-5.4-mini is more expensive but lower-scoring. OAI-4o-mini remains among the lowest-latency models, at 3.84 and 3.93 s per query across the two orders. Second, the anchor-position effect is not specific to

Table 15: Full per-metric delta (anchor-first − corpus-first). Deltas (%) are computed from the unrounded scores. Green indicates improvement when the anchor moves to the start, and red indicates degradation. Row colors indicate capability tier. TD, SR, DB, and MR denote Technical Detail, Section Retrieval, Document Browsing, and Multihop Reasoning.

Method	Tier	Overall Metrics (%)					Per-Question-Type (%)							
		ΔL1	ΔL2	ΔEM	ΔF1	ΔRGL	ΔTD-L1	ΔTD-L2	ΔSR-L1	ΔSR-L2	ΔDB-L1	ΔDB-L2	ΔMR-L1	ΔMR-L2
OAI-4o-mini (backbone)	T0	+12.0	+6.8	+1.1	+3.5	+4.0	+7.6	+9.4	+22.8	+7.5	+15.0	-2.2	+2.5	+4.5
BM25	T1	+10.9	+3.8	+1.1	+2.2	+2.3	+6.3	+5.0	+19.9	-2.6	+14.8	+0.1	+2.2	+7.9
OAI-Emb-3-Large	T1	+7.1	+3.8	-0.4	-0.6	-0.2	+3.5	+3.5	+9.8	+7.7	+17.1	-2.3	+0.5	+1.1
OAI-Emb-3-Small	T1	+9.8	+6.0	+0.8	+0.9	+1.8	+1.5	+4.3	+21.4	+13.8	+10.0	+0.1	+7.6	+0.4
Qwen3-Embed-4B	T1	+9.0	+6.8	+0.4	+1.3	+1.2	+8.6	+4.6	+8.1	+4.8	+12.5	+4.8	+7.1	+12.1
Contriever	T1	+13.5	+8.3	+0.8	+6.1	+6.0	+0.8	+2.8	+25.7	+15.8	+9.9	-2.3	+22.5	+13.5
Cognee	T2	+6.8	+2.3	+0.8	+1.3	+0.7	+0.4	-0.9	+16.8	+1.7	+15.3	-1.7	-3.7	+8.7
HippoRAG-v2	T2	+5.3	+2.3	+4.1	+1.7	+1.8	-5.6	-1.8	+13.5	+6.4	+17.3	+2.8	+3.1	+0.7
GraphRAG	T2	+3.4	+1.9	0.0	-0.1	+0.2	-3.5	-2.1	+2.9	+4.2	+12.3	0.0	+7.4	+4.7
MemoRAG	T2	+18.8	+9.8	+1.1	+6.8	+6.9	+11.2	+7.3	+11.3	+11.7	+45.3	0.0	+20.3	+17.5
RAPTOR	T2	+1.9	+1.5	+3.0	+1.0	+1.1	+5.0	+0.8	+0.1	+5.4	-1.9	0.0	-2.1	-3.3
Self-RAG	T3	+3.0	+5.3	+1.1	+1.9	+2.2	+1.4	+2.3	+2.4	+7.1	+5.0	0.0	+3.6	+10.1
MemPalace-Hybrid	T4	+8.6	+3.4	+1.5	+2.7	+2.9	+6.3	+5.6	+15.4	+3.2	+17.0	+2.4	-5.4	-5.1
MemPalace-Raw	T4	+7.1	+3.8	+1.5	+1.4	+1.4	+4.4	+4.6	+12.1	-1.2	+7.6	+2.4	+1.0	+3.4
REMem	T4	+6.4	+4.5	0.0	+0.6	+0.7	+3.5	+4.6	+17.5	+5.8	+0.7	+5.1	-1.1	-1.7
SimpleMem	T4	-0.8	-2.3	0.0	-2.0	-2.9	-0.4	-4.5	-2.8	-2.1	+15.2	+14.3	-13.5	-13.8
GAM	T4	-1.1	-1.5	0.0	-2.8	-2.2	-2.7	-7.2	+3.1	+3.7	-6.4	0.0	-2.1	-1.8

Table 16: Long-context backbone comparison using the T0 full-context condition on PerDocBench's 266 QA pairs. Prices are OpenRouter list prices in USD per 10^6 tokens from May 2026. **L1** and **L2** are lenient and strict LLM-judge correctness (%). ΔL1 and ΔL2 (%) are anchor-first minus corpus-first, computed from the unrounded scores. **Q(s)** is per-query latency.

Model	Released	Price ($/M tok)		Corpus-first			Anchor-first			Δ (Anchor-first − Corpus-first)	
		In	Out	L1	L2	Q(s)	L1	L2	Q(s)	ΔL1	ΔL2
OAI-4o-mini	2024-07	0.15	0.60	48.1	26.3	3.84	60.2	33.1	3.93	+12.0	+6.8
GDM-2.5-Flash	2025-06	0.30	2.50	14.3	10.2	5.02	20.7	12.8	3.87	+6.4	+2.6
OAI-5.4-mini	2026-03	0.75	4.50	45.5	24.8	6.69	48.1	27.1	3.55	+2.6	+2.3
MiniMax-M2.7	2026-03	0.279	1.20	11.3	4.5	18.08	15.8	8.3	14.25	+4.5	+3.8
DeepSeek-v4-flash	2026-04	0.10	0.20	59.8	38.3	8.47	62.4	42.9	6.83	+2.6	+4.6

OAI-4o-mini. Every tested backbone improves from corpus-first to anchor-first, with ΔL1 ranging from 2.6% to 12.0%.

C.8 Extended Case Studies: Multihop and Document-Browsing

The single anchor case in §5.4, the mass-timber LCA question, demonstrates three failure patterns under one question type: *prior snap*, *paraphrase drift*, and *summary erasure*. Table 17 reports the *extended* anchor case, which adds *adjacency over-recall*: the answer is preserved but unverified extra detail is attached. We then apply the same analysis to two question types where degradation is most pronounced. Multihop Reasoning has the deepest mean L2 deficit relative to the backbone at 4.3%. Document Browsing has the broadest effect, with 13 of 16 methods below the backbone on L2 strict. Outputs in the tables below are excerpted from the per-question judge log; bracketed notes identify omissions or summarize an error message. Parentheticals in the *Failure mode* column tag the observed pattern.

Case B (multihop, persona *ML-Hospital-Readmission*). Question: *"In my ML-Hospital-Readmission project, what prior text-classification techniques do LSTM networks show improvement over?"* Gold = {CONVOLUTIONAL NEURAL NETWORKS, SUPPORT VECTOR MACHINES, RANDOM FORESTS}. The right answer is the *exact* three-item list buried inside one corpus chunk; any paraphrase, addition, or omission is scored wrong on L2 strict.

The same failure modes recur in Table 18. Five methods pass by preserving the verbatim chunk: the T0 backbone, the two OAI embeddings, and the two MemPalace variants. Eight methods (BM25, Contriever, Cognee, RAPTOR, GraphRAG, MemoRAG, Self-RAG, SimpleMem) exhibit *prior snap*:

Table 17: Case A extended (anchor question, corpus-first). The extended version of Table 4; question, gold, and colour conventions identical. Rows grouped by failure cluster: correct → *adjacency over-recall* → *prior snap* → *paraphrase drift* → *summary erasure*.

Method	Tier	Output (verbatim)	Failure mode
OAI-4o-mini (backbone)	T0	"…**cradle-to-construction site**, including the product stage (raw material extraction, processing, transporting, and manufacturing) plus transportation to the construction site."	**correct**
BM25	T1	"…from a **cradle-to-construction site** perspective, including raw material extraction, processing, transporting, and manufacturing, plus transportation to the construction site."	**correct**
HippoRAG-v2	T2	"**Cradle-to-construction site** system boundary."	**correct**
Cognee	T2	"…the **cradle-to-construction site** approach."	**correct**
MemPalace-Raw	T4	"…a **cradle-to-construction site** system boundary, which includes the product stage … plus transportation to the construction site (modules A1–A4)."	**correct**
OAI-Emb-3-Small	T1	"…from **cradle-to-construction site**, *including the product stage (raw material extraction, processing, transporting, and manufacturing) plus transportation to the construction site*."	unsupported extra detail
MemPalace-Hybrid	T4	"…**cradle-to-construction site**… *(modules A1–A4)*."	unsupported extra detail
GraphRAG	T2	"…from **cradle to grave**, covering all life cycle stages from the acquisition of raw material to the disposal."	wrong detail (*prior snap*)
MemoRAG	T2	"…from the acquisition of raw material to the disposal, referred to as the **cradle-to-grave** approach."	wrong detail (*prior snap*)
Self-RAG	T3	"A **cradle-to-grave** LCA approach is defined to compare lifecycle impacts."	wrong detail (*prior snap*)
SimpleMem	T4	"**Cradle-to-grave**."	wrong detail (*prior snap*)
Contriever	T1	"…includes the entire lifecycle from **raw material extraction through processing and manufacturing to end-of-life**…"	wrong detail (*paraphrase drift*)
RAPTOR	T2	"The system boundary for the mass-timber LCA study is **not explicitly defined** in the provided context."	non-answer (*summary erasure*)

they substitute a canonical text-classification baseline list (logistic regression, tree-based methods, Naive Bayes, GLM) for the document's three-item list. The upstream cause differs across systems—in some cases the retriever simply misses the relevant chunk, in others an LLM intermediate step rewrites the candidate list before the backbone—but the diagnostic shape is identical: the answer is the model's training prior, not the document. Three methods (HippoRAG-v2, REMem, GAM) exhibit *summary erasure*: the offline memory architecture cannot surface the gold list at query time and the system reports a refusal. One method, Qwen3-Embed-4B, exhibits *adjacency over-recall*, extending the gold list with *decision tree learning*—a near-but-non-gold neighbour pulled in by a higher-dimensional encoder.

Case C (Document Browsing, persona *SmartCity-Blockchain*). Question: *"In my SmartCity-Blockchain project, considering the challenges we're facing with urban data integration, what papers do I have on blockchain integration?"* Gold = {*Advanced IoT–Blockchain Integration…*, *A Blockchain–Artificial Intelligence Convergence Framework…*}—two real corpus titles. This is a list-mode retrieval question, the type with the broadest aggregate degradation footprint. A total of 13 of 16 memory methods fall below OAI-4o-mini on Document Browsing L2 strict in Table 13.

The list-mode failure pattern in Table 19 is sharper, and a fifth cluster appears. Only four systems name both gold papers: the backbone, HippoRAG-v2, Cognee, and REMem. Five systems show *partial list recall*: OAI-Emb-3-Small, OAI-Emb-3-Large, Qwen3-Embed-4B, SimpleMem, and GAM name the first paper but omit the second. Two systems, RAPTOR and MemoRAG, show *summary erasure* by refusing despite both papers being in the corpus. Contriever shows *paraphrase drift* by retrieving an unrelated paper. The remaining five systems show *pipeline failure*. BM25, MemPalace-Hybrid, and MemPalace-Raw generate a header but truncate before listing any title. GraphRAG returns only a graph-node hash, and Self-RAG returns a content-policy error.

Cross-case takeaway. Across the two additional cases, the four surface patterns recur with different distributions. Case B shows prior snap, summary erasure, and adjacency over-recall. Case C shows paraphrase drift, summary erasure, and partial list recall. Pipeline failure appears only in the list-mode Document Browsing case. Methods near the bottom of Table 3 also exhibit visible failures here.

Table 18: Case B (multihop, corpus-first). Verbatim outputs for the ML-Hospital-Readmission case. **Green** marks the gold three-item list when present. **Red** marks canonical text-classification baselines that replace the document-specific list. Five methods recover all three items, one adds an extra item, eight substitute a canonical list, and three refuse.

Method	Tier	Output (verbatim)	Failure mode
OAI-4o-mini	T0	"…**convolutional neural networks, support vector machines, and random forests**."	**correct**
OAI-Emb-3-Small	T1	"…**convolutional neural networks, support vector machines, and random forests**."	**correct**
OAI-Emb-3-Large	T1	"…**convolutional neural networks, support vector machines, and random forests**."	**correct**
MemPalace-Hybrid	T4	"…**convolutional neural networks, support vector machines, and random forests** in text classification tasks."	**correct**
MemPalace-Raw	T4	"…**convolutional neural networks, support vector machines, and random forests** in text classification tasks."	**correct**
Qwen3-Embed-4B	T1	"**decision tree learning**, convolutional neural networks, support vector machines, and random forests."	unsupported extra item (*adjacency over-recall*)
BM25	T1	"…**traditional regression** and other ML methods, including **tree-based methods** and SVM."	wrong detail; missing gold item (*prior snap*)
Contriever	T1	"…**logistic regression and generalized linear models (GLM)**…"	wrong detail; missing gold item (*prior snap*)
Cognee	T2	"…**tree-based methods, neural networks, regularized logistic regression**, and SVM…"	wrong detail with extra item (*prior snap*)
RAPTOR	T2	"**Regularized logistic regression, decision trees, and SVM**."	wrong detail; missing gold item (*prior snap*)
GraphRAG	T2	"…**Naive Bayes** and SVM."	wrong detail; missing gold item (*prior snap*)
MemoRAG	T2	"…**logistic regression**, SVM, and other basic ML algorithms."	wrong detail; missing gold item (*prior snap*)
Self-RAG	T3	"SVM and **other traditional regression and ML models**…"	too generic; missing gold item (*prior snap*)
SimpleMem	T4	"**Logistic regression and tree-based methods**."	wrong detail; missing gold item (*prior snap*)
HippoRAG-v2	T2	"**The provided user memory does not specify…**"	non-answer (*summary erasure*)
REMem	T4	"**The snippets do not provide specific information…**"	non-answer (*summary erasure*)
GAM	T4	"**…not found in the available content.**"	non-answer (*summary erasure*)

SimpleMem fails through prior snap or incomplete recall, while Self-RAG and GraphRAG show prior snap or pipeline failure. These cases illustrate distinct ways to produce an incorrect answer; they do not establish how often each mechanism accounts for the aggregate result.

C.9 Answer-String Matching and Outcome Analysis

The case studies above inspect individual failures. This subsection compares answer correctness with lexical matches to gold-answer strings in the recorded generator input. It provides an aggregate diagnostic without treating a missing string as proof of evidence loss at a particular stage.

Protocol. For every method whose retrieval output can be matched to individual queries, we log per-query intermediate records {retrieval query, retrieved passages, final answer} during one of the three main corpus-first runs and join them to the per-query judge verdicts. Table 13 reports a different corpus-first run, so its correct column need not match this analysis. We cover 15 of the 16 memory methods. The T0 backbone has no retrieval stage, and GAM's agentic filesystem loop exposes no passages that can be aligned to queries, so neither is included. GraphRAG contributes 250 instances because 16 queries have no alignable artifact; each other method contributes 266.

Matching rule and scope. We lowercase gold strings and the concatenated retrieved text, remove punctuation and the articles *a*, *an*, and *the*, and normalize whitespace. A match occurs when any nonempty normalized gold string is a substring of that text. Lists of gold strings are flattened and tested with an *any-match* rule; a match therefore need not cover every item of a list answer. The rule misses semantic equivalents with different wording and does not verify that a matched string appears in the correct context. Both matched and unmatched instances can be answered correctly.

Table 19: Case C (Document Browsing, corpus-first). Outputs for the SmartCity-Blockchain Document Browsing case. Bracketed text summarizes the error message. **Green** marks the gold paper title when correctly named; **red** marks empty headers, hash-only outputs, content-policy trips, and outright refusals. Four methods name both papers; five name only the first; eight fail to deliver any usable title.

Method	Tier	Output (excerpts, paper titles abbreviated)	Failure mode
OAI-4o-mini	T0	"**Advanced IoT–Blockchain Integration…** and **A BLOCKCHAIN–ARTIFICIAL INTELLIGENCE CONVERGENCE FRAMEWORK…**"	**correct** (both papers)
HippoRAG-v2	T2	"**Advanced IoT–Blockchain Integration…** and **A BLOCKCHAIN–ARTIFICIAL INTELLIGENCE CONVERGENCE FRAMEWORK…**"	**correct** (both papers)
Cognee	T2	"1. **Advanced IoT–Blockchain Integration…** 2. **A Blockchain–Artificial Intelligence Convergence Framework…**"	**correct** (both papers)
REMem	T4	"**A BLOCKCHAIN–ARTIFICIAL INTELLIGENCE CONVERGENCE FRAMEWORK…** and **Advanced IoT–Blockchain Integration…**"	**correct** (both papers)
OAI-Emb-3-Small	T1	"1. Advanced IoT–Blockchain Integration… **[paper B missing]**"	missing gold item (*partial list recall*)
OAI-Emb-3-Large	T1	"1. Advanced IoT–Blockchain Integration… **[paper B missing]**"	missing gold item (*partial list recall*)
Qwen3-Embed-4B	T1	"1. Advanced IoT–Blockchain Integration… **[paper B missing]**"	missing gold item (*partial list recall*)
SimpleMem	T4	"Paper titled 'Advanced IoT–Blockchain Integration…' **[paper B missing]**"	missing gold item (*partial list recall*)
GAM	T4	"1. Advanced IoT–Blockchain Integration… *[verbose paragraph; paper B never named]*"	missing gold item (*partial list recall*)
BM25	T1	"**You have the following papers on blockchain integration:**" *[output truncated, no titles]*	non-answer (*pipeline failure*)
Contriever	T1	"**The Importance of System Engineering Competences…** [unrelated]"	non-answer (*paraphrase drift*)
RAPTOR	T2	"**The context provided does not contain any specific papers…**"	non-answer (*summary erasure*)
GraphRAG	T2	"**Paper ID: 85da9a47591779419dc2a87fb48c0bf7d0bca6e0**" *[hash, no title]*	non-answer (*pipeline failure*)
MemoRAG	T2	"**You do not have any specific papers on blockchain integration…**"	non-answer (*summary erasure*)
Self-RAG	T3	"**Error code: 400… [content-policy error]…**"	non-answer (*pipeline failure*)
MemPalace-Hybrid	T4	"**You have two relevant papers on blockchain integration:**" *[output truncated, no titles]*	non-answer (*pipeline failure*)
MemPalace-Raw	T4	"**You have the following papers on blockchain integration:**" *[output truncated, no titles]*	non-answer (*pipeline failure*)

We compute match rates and conditional correctness from the match indicator for *all* instances, independently of the outcome categories below.

Outcome categories. Each instance receives one of four mutually exclusive labels in Figure 4 and Table 21. **Correct** includes every L1-correct answer, regardless of matching. Among incorrect answers, **No match (passage)** and **No match (memory)** distinguish passage-output pipelines from intermediate-memory pipelines; **Match, incorrect** includes either family when a gold string matches. The family split describes the system interface, not a localized failure mechanism. In particular, no match in an intermediate-memory output does not distinguish a construction omission from a retrieval miss, and no match in either family does not establish that supporting evidence is absent.

Results. Across 3,974 instances, the four categories contain 1,746 correct answers (43.9%), 745 incorrect answers with no match in passage-output pipelines (18.7%), 1,200 incorrect answers with no match in intermediate-memory pipelines (30.2%), and 283 matched but incorrect answers (7.1%). Thus, 1,945 of the 2,228 incorrect answers (87.3%) have no match. Table 20 cross-tabulates matching and correctness for all instances: 772 have a match, of which 489 are correct (63.3%), whereas 1,257 of 3,202 unmatched instances are correct (39.3%).

Pipeline-family comparison. The six passage-output methods contribute 1,596 instances and 796 correct answers. Of their 167 matched instances (10.5%), 112 are correct (67.1%). The nine intermediate-memory methods contribute 2,378 instances and 950 correct answers. Of their 605 matched instances (25.4%), 377 are correct (62.3%). Architecture, query policy, and match rates

Table 20: Answer-string matching and L1 correctness in one corpus-first run. Counts cover the same 3,974 instances as Table 21. Matching is evaluated independently of correctness.

	L1 correct	L1 incorrect	Total
Match	489	283	772
No match	1,257	1,945	3,202
Total	1,746	2,228	3,974

vary between the groups, so the comparison does not isolate the effect of constructing an intermediate memory. The per-method outcome percentages appear in Table 21.

Table 21: Answer outcomes in one corpus-first run. Entries are within-row percentages. Correct includes all L1-correct answers. The other columns split incorrect answers by answer-string matching and, for unmatched instances, pipeline family. A missing match does not establish missing evidence or localize an error to construction or retrieval. GraphRAG has N=250 because 16 queries produced no alignable artifact. Methods are sorted by correct rate. The final three rows correspond to Figure 4; row colors indicate capability tier.

Method	Tier	N	Correct	No match (passage)	No match (memory)	Match, incorrect
BM25	T1	266	54.1	43.2	—	2.6
OAI-Emb-3-Large	T1	266	52.3	45.1	—	2.6
Qwen3-Embed-4B	T1	266	50.8	45.5	—	3.8
MemPalace-Hybrid	T4	266	50.8	—	38.7	10.5
OAI-Emb-3-Small	T1	266	50.4	46.2	—	3.4
MemPalace-Raw	T4	266	49.6	—	41.4	9.0
HippoRAG-v2	T2	266	46.6	49.2	—	4.1
Contriever	T1	266	45.1	50.8	—	4.1
REMem	T4	266	44.7	—	50.0	5.3
MemoRAG	T2	266	44.4	—	50.8	4.9
Cognee	T2	266	38.0	—	54.1	7.9
Self-RAG	T3	266	36.5	—	49.6	13.9
RAPTOR	T2	266	33.1	—	51.1	15.8
GraphRAG	T2	250	31.6	—	63.2	5.2
SimpleMem	T4	266	30.5	—	56.0	13.5
Aggregate summary						
All analyzed methods (15)	—	3,974	43.9	18.7	30.2	7.1
Passage-output pipelines (6)	—	1,596	49.9	46.7	—	3.4
Intermediate-memory pipelines (9)	—	2,378	39.9	—	50.5	9.6

Anchor case revisited. The records for Case A (§5.4) provide more specific evidence than aggregate matching alone. Contriever retrieves the wrong detail, while GraphRAG and MemoRAG alter or omit the qualifier in their intermediate outputs. Self-RAG retains the gold span in four retrieved passages, yet its final answer changes the boundary to the more familiar *cradle-to-grave* scope. Every other method in this case whose retrieved evidence contains the gold span answers the question correctly.

D PROMPTS

This appendix collects prompts used in PerDocBench's data construction, anchor-conversation generation, cross-source rewriting, answer judging, and distractor-rewrite attacks. Prompts are colour-coded by category: **blue** for dataset construction (corpus, question, schema), **green** for anchor-conversation generation and cross-source rewriting, **purple** for LLM judges, and **red** for the distractor-rewrite attack. When a prompt has no custom system message, the LLM client default

`"You are a helpful assistant."` is used and we show only the user template; otherwise both **System** and **User** are listed.

D.1 Dataset Construction Prompts

Persona / discipline-tree match

System. You are a helpful assistant that analyzes academic disciplines and matches them to research interests.

User. You are matching academic disciplines to a researcher's interests. Given the following persona description, carefully identify the top `{top_k}` most relevant academic disciplines from the list below.
IMPORTANT: Consider the specific research topic, methodology, subject matter, and geographic/cultural focus mentioned in the persona. Be precise and avoid overly general disciplines unless they are truly the best match.
Persona:
`{persona_text}`
Academic Disciplines:
`{disciplines_str}`
Instructions:
1. Analyze the persona's research focus carefully.
2. Match disciplines that directly relate to the specific topic, methodology, or subject area.
3. Prioritize more specific disciplines over general ones when both are relevant.
4. Consider interdisciplinary connections if mentioned.
Please respond with ONLY the numbers of the top `{top_k}` most relevant disciplines, separated by commas (e.g., "1, 3, 5"). If you think fewer than `{top_k}` disciplines are relevant, you can return fewer numbers.

Persona / topic re-ranker

System. You are a helpful assistant that analyzes academic disciplines and matches them to research interests.

User. You are ranking academic research topics by relevance to a researcher's interests. Given the following persona description, rank the research topics from most relevant to least relevant.
Persona:
`{persona_text}`
Research Topics:
`{topics_str}`
Please respond with:
1. The numbers of the topics in order of relevance (most relevant first), separated by commas.
2. For each topic, provide a relevance score from 0.0 to 1.0 (where 1.0 is highly relevant).
Format your response as:
Ranking: 3, 1, 5, 2, 4
Scores: 3:0.95, 1:0.90, 5:0.85, 2:0.75, 4:0.60

Semantic-Scholar query generation

User. Based on this researcher persona:
`{persona}`
And these topics: `{topics}`
Generate `{n}` specific, high-quality academic search queries to find foundational and recent papers. Return ONLY a JSON list of strings, e.g. `["query1", "query2"]`.

Per-paper topic extraction (used by `document_browsing`)

User. Extract 2–3 main research topics from this paper.
Title: `{title}`
Abstract: `{abstract}`
Return JSON: `{"topics": ["topic1", "topic2"]}`. Topics should be concise (2–4 words each), like "soil moisture prediction", "plant disease detection".

document_browsing question generator

User. Generate a natural question asking about documents in a personal library.
Topic: `{topic}`
Example papers on this topic: `{paper_titles}`
The question should:
1. Ask what papers/documents the user has on this topic.
2. Sound natural, like asking a personal assistant.
3. NOT mention specific paper titles.
4. Be under 20 words.
Examples:
- "What papers do I have about machine learning for crop yield prediction?"
- "Do I have any documents on computer vision in agriculture?"
- "Which studies in my library discuss soil moisture estimation?"
Return JSON: `{"question": "...", "filter_topic": "the topic keyword for filtering"}`.

section_retrieval question generator

User. Generate a question that requires looking at a specific section of a paper.
Paper Title: `{title}`
Abstract: `{abstract}`
Target Section: `{section}`
Detail types in this section: `{detail_types}`
Create a question that:
1. Describes the paper by its KEY CHARACTERISTICS (not the full title) — use phrases like "the paper about…", "the study on…", "the research that proposes…".
2. Asks about something typically found in the `{section}` section.
3. Has a CONCRETE answer (not vague).
4. Is under 30 words.
Good examples:
- "In the paper about 2D CNN for soil moisture prediction, what loss function is used according to the methods?"
- "What accuracy does the rice yield prediction model achieve in the results section?"
- "According to the study on plant disease detection with deep learning, what dataset is mentioned in the methods?"
Return JSON: `{"question": "...", "paper_description": "...", "answer": "...", "explanation": "...", "evidence_snippet": "...", "section": "{section}"}`.

technical_detail question generator

User. You are creating a factual question based on a knowledge graph triplet.
Triplet: `"{subj}" -[{rel}]-> "{obj}"`
Source Text (where this fact comes from):
`{context}`
Your task: Create a SHORT question where the answer is `"{obj}"`.
Requirements:
1. Question should be under 25 words.
2. Question should directly ask for `"{obj}"` as the answer.
3. Use simple, natural language.
4. The question must be answerable from the source text.
Examples of GOOD questions:
- For `(BERT, EVALUATES_ON, GLUE)`: "What benchmark is used to evaluate BERT?" → "GLUE".
- For `(ResNet, USES_METHOD, skip connections)`: "What technique does ResNet use to enable deeper networks?" → "skip connections".
Examples of BAD questions:
- "In the context of natural language processing advancements…" (too wordy).
- "What innovative approach…" (vague phrasing).
Return JSON: `{"question": "...", "answer": "{obj}", "explanation": "...", "evidence_snippet": "..."}`.

multihop_reasoning question generator

User. You are creating a multi-hop reasoning question based on a knowledge graph path.
Knowledge Path:
- Step 1: "{subj}" -[{rel1}]-> "{mid}"
- Step 2: "{mid}" -[{rel2}]-> "{obj}"
The answer "{obj}" is a {answer_type} (based on relation type "{rel2}").
Source Text:
{context}
Your task: Create a SHORT question where the answer is "{obj}".
CRITICAL: The answer is a {answer_type}, so ask "what {answer_type}..." accordingly:
- MEASURES_WITH → ask "what metric..." (NOT "what method...").
- USES_METHOD → ask "what method...".
- EVALUATES_ON → ask "what dataset...".
Requirements:
1. Question under 25 words.
2. Ask for the correct type ({answer_type}).
3. Answer must be "{obj}".
Return JSON: {"question": "...", "answer": "{obj}", "explanation": "...", "evidence_snippet": "..."}.

D.2 Anchor-Conversation and Cross-Source Prompts

Anchor assess-and-clean

User. You are cleaning and screening a candidate anchor chunk for a RAG benchmark.
Return JSON only with the following schema:
{"keep": true/false, "cleaned_text": "...", "reason": "...", "project_name": "...", "keywords": ["..."]}
Guidelines:
- If the chunk is mostly references, author lists, tables/figures, page headers, OCR noise, or non-English text, set keep=false and cleaned_text="".
- Otherwise, clean the text into a coherent English paragraph (remove references, page numbers, OCR artifacts).
- Project name should be a short codename (1–3 words, letters/numbers/hyphen), topic-aligned, not the paper title.
- Keywords should be 3–6 topical English terms drawn from the cleaned_text.
Chunk:
{snippet}

Conversation segment generation (4–6 turns)

User. You are a synthetic data generator for evaluating RAG systems.
Persona:
{persona}
Timeframe:
{time_context}
Anchor Knowledge (excerpt from a paper the user has in their library):
Title: {anchor_paper_title}
Excerpt: {anchor_text}
Instructions:
1) Create a multi-turn chat log between a researcher (User) and an AI assistant.
2) The user is working on a concrete project with a codename: {project_name}. Use this exact name consistently.
3) The user describes their setup, results, or confusion in a way that is subtly connected to the Anchor Knowledge.
4) The chat should reflect the Timeframe context (progressively more mature from init to next_year).
5) The chat should NOT mention the paper title or quote the excerpt.
6) The chat should be background context only; do NOT include direct benchmark-style questions.
7) Keep it natural and slightly technical. Avoid bullet points.
8) 4–6 exchanges total. STRICTLY alternate User/Assistant turns.
9) Required keywords: {keyword_line}
10) STRICT: Mention at least 2 of the required keywords verbatim.

Output JSON only:
`{"project_name": "...", "conversation": ["User: ...", "Assistant: ...", "..."], "event_description": "..."}`

Reflection / continuity pass

User. You are refining a multi-turn chat log so segments flow as one session.
Return JSON only:
`{"conversation": ["User: ...", "Assistant: ...", "..."]}`
Rules:
1) Rewrite ONLY the `NEW_SEGMENT` so it continues after `PREVIOUS`.
2) Do NOT start with greetings like "Hi/Hello/Hey".
3) Avoid phrases like "I've just started" unless this is the very beginning (it is not).
4) Do NOT introduce a different project; keep the same project throughout.
5) Keep speaker order and turn count the same.
6) Preserve any existing `Side_Note` line and its timestamp if present.
7) Keep the exact project codename if mentioned: `{project_name}`.
PREVIOUS:
`{prev_snippet}`
NEW_SEGMENT:
`{next_segment}`

Cross-source rewrite (chat-stem ⊕ doc-Q)

User. You are a synthetic data generator for evaluating RAG systems.
Persona:
`{persona}`
Anchor Conversation Context (background only):
`{convo_text}`
Anchor Event Description:
`{event_desc}`
Anchor Knowledge Excerpt (for grounding, do NOT quote directly):
`{anchor_excerpt}`
Benchmark Question (do NOT restate directly in output):
`{q_text}`
Ground-truth Answer (for reference only):
`{answer}`
Ground-truth Explanation (for reference only):
`{answer_explanation}`
Instructions:
1) Write ONE natural user question that smoothly connects anchor context with the benchmark intent.
2) The question MUST preserve benchmark semantic target and information need; only rephrase style.
3) The question should sound conversational, not templated.
4) Avoid rigid openers like "Based on our discussion", "According to our discussion", or direct prefix stitching.
5) Use user-centric possessives/pronouns: prefer "my", "the user's", or the user's name; do NOT use "your" to refer to the user's project/library.
6) Do NOT mention paper titles or quote anchor excerpts.
7) Keep it concise and slightly technical.
8) Provide an explicit `required_query_rewrite` for document search.
9) Preserve source gold semantics: do NOT change gold paper mapping intent from benchmark.
10) Keep the evidence snippet short (one sentence) and grounded in anchor context.
11) INTERNAL PLANNING (do not output): first draft three micro-parts in your head: (a) anchor callback phrase, (b) transition phrase, (c) benchmark ask core. Then merge them into ONE fluent question sentence.
12) The final question must include at least one transition connector, such as: "given that", "in that context", "building on that", "from that project context".
13) If the question contains anchor context but no transition connector, rewrite it once to add a natural transition.
`{task_specific_clauses}`
Good example:
- "In my bio-scaffold-review project, building on our earlier discussion of neural scaffold design, what analysis technique is typically used to study 3D helical coils?"

Bad example (too abrupt, no transition):
- "What analysis technique is typically applied for studying 3D helical coils?"
Output JSON only:
`{"question": "...", "required_query_rewrite": "...", "ground_truth": "...", "ground_truth_explanation": "...", "evidence_snippet": "..."}`

D.3 LLM-JUDGE PROMPTS

Data-quality four-axis audit (judge: DeepSeek-v4-flash)

System. You are a strict dataset quality auditor for long-memory QA data. Score each axis independently, require evidence rather than plausibility, and return only JSON.

User. Score the item below along four 0/1/2 axes:
`semantic_evidence_support`: 0 unsupported, 1 partially supported or inferential, 2 directly supported by the evidence snippets.
`answerability`: 0 ambiguous/unanswerable, 1 answerable with caveats, 2 clear and uniquely answerable.
`context_necessity`: 0 answerable without the memory context, 1 weakly context-dependent, 2 requires the provided long-memory context.
`shortcut_resistance`: 0 obvious answer leak or keyword shortcut, 1 some shortcut risk, 2 robust against shallow shortcuts.
Required output schema:
`{"scores": {<dim>: 0|1|2}, "pass": bool, "issues": ["unsupported_answer", "ambiguous_question", "weak_context_need", "shortcut_risk", ...], "rationale": "one concise sentence"}`.
Set `pass=true` only if the critical dimensions `semantic_evidence_support` and `answerability` are both ≥ 1 and the total score is $\geq 6/8$.
Item:
`{"dataset": ..., "variant": ..., "question_type": ..., "question": ..., "answer": ...}`
Evidence snippets:
`{evidence_snippets}`

L1 / L2-strict answer judge (judge: OAI-4o)

System. You are an impartial judge for a research benchmark on long-context personal memory retrieval. Judge conservatively. Use only the provided question, gold answer, task metadata, and model response. Do not use external world knowledge to rescue an answer. Return valid JSON only.

User. Benchmark: PerDocBench
Prompt version: `{prompt_version}`
Question type: `{question_type}`
Task description:
`{task_description}`
Scoring rubric:
1. `{level_1_rubric}`
2. `{level_2_rubric}`
3. If the response is empty, refuses to answer, or clearly does not answer the question, both levels must be 0 and `error_tags` should include `non_answer`.
4. If the response adds unsupported specifics or fabricated titles/details, include `extra_unsupported_detail` or `hallucinated_document` when appropriate.
5. Use only these error tags when needed: `{allowed_error_tags}`.
Original question:
`{question}`
Question rewrite hint:
`{rewrite_text}`
Expected retrieval/tool pattern:
`{tool_call_text}`
Gold answer:
`{gold_answer}`
Model response:
`{hypothesis}`
Return JSON only in this exact schema:
`{"level_1_correct": 0, "level_2_strict": 0, "error_tags": [], "brief_rationale": "short explanation"}`.

Per-question-type rubric strings (filled into the L1/L2 judge above)

`technical_detail`.
Task. The question asks for a precise technical fact, such as a method name, dataset, metric, entity, location, number, or short phrase.
L1. Set `level_1_correct` to 1 if the response contains the correct requested detail or an unambiguous equivalent. Minor wording variation is allowed. Set it to 0 if the key detail is wrong, missing, or too generic.
L2. Set `level_2_strict` to 1 only if the response is precise, directly answers the requested detail, and does not add unsupported or conflicting technical content.

`section_retrieval`.
Task. The question targets a specific document section such as methods, results, or discussion and expects the relevant section-specific finding or method.
L1. Set `level_1_correct` to 1 if the response recovers the core requested section content or a semantically equivalent statement.
L2. Set `level_2_strict` to 1 only if the response stays focused on the requested section target and does not drift into unsupported or unrelated content.

`document_browsing`.
Task. The question asks which documents or papers are relevant in the user's document library. The gold answer may be one title or a short list of titles.
L1. Set `level_1_correct` to 1 if the response identifies at least the main correct document(s). Minor title shortening or formatting changes are acceptable when the document is still uniquely identifiable.
L2. Set `level_2_strict` to 1 only if the response has strong coverage of the gold title set and does not introduce hallucinated or clearly irrelevant documents. For 1–2 gold titles, require all of them. For 3 or more gold titles, allow at most one omission.

`multihop_reasoning`.
Task. The question requires combining multiple clues from the user's memory to produce a final conclusion.
L1. Set `level_1_correct` to 1 if the final answer is correct or clearly equivalent, even if the wording differs.
L2. Set `level_2_strict` to 1 only if the response gives the correct final answer and does not rely on unsupported leaps, contradictions, or generic topic talk instead of the resolved answer.

D.4 Distractor-Rewrite Attack Prompt

Distractor-rewrite (answer-preserving density attack)

User. Rewrite the evidence utterance into a longer, natural conversational utterance with lower information density. Preserve the exact fact needed to answer the question and do not introduce contradictions. Do not quote the original evidence verbatim; rephrase it while keeping the answer supported. Bury the answer-supporting fact among plausible but irrelevant surrounding details, side comments, and background noise. Add 2–3 nearby distractor details such as another time, place, name, plan, or activity, but make them clearly unrelated side details rather than contradictions. Do not use phrases that highlight the evidence, such as "the important part" or "the key fact". Do not repeat the gold answer more times than it appeared in the original evidence. Return JSON only with keys: `rewritten_evidence`, `preserved_fact`, `answer_still_supported`.
Question: `{question}`
Gold answer: `{answer}`
Original evidence: `{original}`

Companion question-rewrite (combined attack variants)

User. Rewrite the question to ask for exactly the same answer, but use a different surface form. Do not include the answer. Return only the rewritten question.
Question: `{question}`
Answer: `{answer}`

E LIMITATIONS

Our controlled evaluation has several limits. Main comparisons use OAI-4o-mini; Appendix C.7 tests source order with four additional in-context backbones, not the transfer of all memory methods across backbones. Ten scholarly personas from one source provide limited coverage of professions, personal domains, and source types. The common two-stage interface also omits method-specific tuning. Broader data and tailored implementations could help distinguish limitations of memory design from those of this evaluation protocol. Finally, lexical answer matching can miss paraphrases and does not establish the cause of individual errors.