Grove on LongMemEval

Session-level recall@5 and recall@10 on the public benchmark for long-term memory in AI agents.

Author Grove

Date 1 June 2026

Dataset longmemeval_s_cleaned · n = 470

ABSTRACT

We report Grove's session-level recall@5 and recall@10 on the public LongMemEval benchmark[1], the standard evaluation for long-term memory in conversational AI agents. The benchmark measures whether a system, given a long multi-session chat history, can recover the specific past sessions that hold the evidence needed to answer a question. On the cleaned _s split, which contains 470 scored questions, each set against a haystack of roughly 40 candidate sessions, Grove surfaces a correct evidence session within its five highest-ranked results on 99.36% of questions, and within its ten highest-ranked results on 99.79%. The R@5 figure exceeds the published scores of gbrain's hybrid configuration (97.60%)[2], gbrain's vector-only configuration (97.40%)[2], and MemPalace's raw baseline (96.60%)[3]. No comparator publishes a recall@10 figure. Every score in this paper is produced by LongMemEval's own retrieval scorer, imported and run without modification, so the comparison rests on the benchmark authors' definition of correctness rather than our own. The only departure from a plain dense retriever is an index-time fact-extraction step paired with a two-arm hybrid ranking, and we isolate the contribution of that step in a controlled ablation. The sections that follow describe the dataset, the metric and its exact definition, the retrieval pipeline and the reasoning behind it, the ablation, the provenance of each comparator number, and a complete reproduction path. We close by stating which categories of memory-system behavior the benchmark does not exercise, so that the headline number is not read as a claim it cannot support.

1 Background

A long-running AI agent accumulates a growing record of sessions, decisions, and user-stated facts over days and weeks of interaction. Once that record outgrows any usable context window, the central engineering problem becomes retrieval: given a new question, which fragments of the accumulated history should be pulled back into context so that the agent can answer correctly? To a first approximation, the quality of an agent's memory is the quality of this retrieval step, and it is the part of a memory system that can be measured without confounding it with the behavior of whatever model later reads the result.

LongMemEval[1] is, to our knowledge, the only public benchmark that scores this retrieval step directly and in isolation. It consists of 500 questions, each paired with a long multi-session chat history. For every question the benchmark marks one or more *gold evidence sessions*, the sessions that actually contain the information needed to answer (recorded in the `answer_session_ids` field), and surrounds them with approximately 40 distractor sessions on unrelated topics. A system is scored on how well it separates the former from the latter.

The benchmark defines two evaluation tracks. The *retrieval track* is deterministic: it scores the ranked list of sessions a system returns, using recall@k and NDCG@k, with no model anywhere in the scoring path. The *end-to-end QA track* instead feeds the retrieved context to a reader model and grades the final answer with an LLM judge. This paper concerns only the retrieval track, both because it isolates memory retrieval from answer generation and because it is fully reproducible without a judge in the loop. We report Grove against the two systems most frequently cited on LongMemEval in the personal-knowledge-graph space: gbrain, at 97.60% (hybrid) and 97.40% (vector-only)[2], and MemPalace, at 96.60% (raw)[3].

2 Method

2.1 Dataset

We evaluate on `longmemeval_s_cleaned.json`, the cleaned version of the small (`_s`) split and the current official release on Hugging Face[4]. The file holds 500 questions. Thirty of these are abstention instances, marked with the `_abs` suffix in their identifiers; they test whether a system correctly declines to answer when no supporting evidence exists, which is a property of the reader rather than the retriever, and the official harness excludes them from retrieval scoring. That leaves 470 scored questions.

Each question is accompanied by a chat history of approximately 40 sessions, where a session is one continuous conversation of alternating user and assistant turns. One or more sessions in that history are the gold evidence; the rest are topical distractors, chosen to be plausibly related but not actually answer-bearing. The questions are tagged with six types that stress different retrieval demands: single-session questions grounded in a user turn, an assistant turn, or a stated preference; multi-session questions whose evidence is spread across several conversations; knowledge-update questions in which a later session supersedes an earlier fact; and temporal-reasoning questions that turn on when something was said.

2.2 Metric

The headline metric is session-level `recall_any@5`. For a single question it poses a yes-or-no test: among the five sessions the retriever ranks highest, is at least one of the gold evidence sessions present? The benchmark score is the fraction of the 470 questions for which the answer is yes. Formally, averaging over the non-abstention questions:

$$R@5 \;=\; (1/|Q|) \;\; \Sigma_{q \in Q} \;\; \mathbb{1}[\, \exists s \in G_q \;:\; \mathrm{rank}(s) \le 5 \,] \tag{1}$$

where G_q is the set of gold evidence sessions for question q (its `answer_session_ids` field), *rank(s)* is the position of session s in the retriever's output for that question, Q is the set of 470 non-abstention questions, and the indicator function contributes 1 when at least one gold session lands within the top five and 0 otherwise.

Two properties of this metric matter for reading the results. First, it is session-level and permissive at the question: a question counts as solved as soon as one correct session appears in the top five, regardless of how many gold sessions exist or where the others rank. Second, and more important for reproducibility, the scoring path contains no model. The metric is computed by LongMemEval's own `eval_utils.evaluate_retrieval` function[5], imported and executed unmodified; underneath, it is a deterministic sort of similarity scores followed by a set-membership test in NumPy. There is no LLM judge, and therefore no run-to-run variance and no grader disagreement to absorb. LongMemEval does define a separate end-to-end QA-accuracy metric that relies on a judge, but we do not run that track here.

2.3 Pipeline

For each question the retriever sees only that question's own haystack of roughly 40 sessions and must rank those sessions by relevance to the question. Grove does this in two stages: an index-time stage that distills each session into facts, and a query-time stage that ranks sessions using both those facts and the raw session text.

Stage 1. Fact extraction (index-time)

Each unique session transcript is sent once to `gpt-4.1-mini` with a system prompt that asks for a JSON array of atomic, self-contained, third-person statements about the user: the preferences, attributes, decisions, plans, and events the user reported in that session. The motivation is that a raw transcript interleaves a small amount of durable signal, a few facts about the user, with a large amount of conversational filler, and a short list of clean facts is a far sharper match for a pointed question than the full back-and-forth. Sessions that contain no user-stated facts, which is common for generic assistant chatter, return an empty array and contribute nothing to the fact index. The exact prompt is reproduced in Appendix A.

Stage 2. Hybrid retrieval (query-time)

The question, together with every fact extracted from every session in the haystack, is embedded with `text-embedding-3-large` at 1536 dimensions (set through the `dimensions` parameter) and L2-normalized. The full text of each session is embedded with the same model. Each session then receives two similarity scores against the question: a *fact-arm* score, the highest cosine similarity between the question and any fact drawn from that session, and a *session-arm* score, the cosine similarity between the question and the session's full text. The session's final score is the larger of the two. Taking the maximum, rather than a weighted

sum, lets each session win on whichever representation serves it best: a session whose relevance is captured by one sharp fact is ranked on that fact, while a session whose relevance is diffuse across the conversation is ranked on its full text. Sessions are sorted by this combined score, and the five highest are handed to the scorer.

2.4 Models

Two OpenAI models are used, one per stage, summarized in Table 1. We deliberately match the embedding model and dimensionality that gbrain reports for its vector arm. Embedding strength is the largest single confound in a benchmark of this kind, so holding it fixed means any difference between Grove and gbrain's vector configuration is attributable to the pipeline built around the embedder rather than to the embedder itself.

Table 1. Models used in the Grove pipeline. The embedding model and dimensionality match the configuration gbrain reports for its vector arm, controlling for embedding strength as a confounding variable.

Role	Model	Provider	Configuration
Fact extraction	`gpt-4.1-mini`	OpenAI	API default at run time, temperature 0
Embedding (queries, facts, sessions)	`text-embedding-3-large`	OpenAI	1536 dimensions via `dimensions` parameter, L2-normalized

2.5 Reproducibility

The evaluation pipeline is three short Python scripts whose only dependencies are `openai`, `numpy`, `tiktoken`, and LongMemEval's own scorer, imported unmodified. The full run, from raw dataset to headline number, is reproduced with:

```
# 1. Download the dataset (~265 MB)
wget https://huggingface.co/datasets/xiaowu0162/longmemeval-
cleaned/resolve/main/longmemeval_s_cleaned.json

# 2. Stage 1: distill facts per unique session
#    (gpt-4.1-mini, ~$1.80, ~40 min, 16-way parallel)
python extract_facts.py \
    --in_file  longmemeval_s_cleaned.json \
    --out_file facts/all_facts.jsonl \
    --workers  16

# 3. Stage 2: hybrid retrieval
#    (text-embedding-3-large @ 1536, ~$0.40, ~13 min)
python run_hybrid.py \
    --in_file    longmemeval_s_cleaned.json \
    --facts_file facts/all_facts.jsonl \
    --out_file   out/full500_hybrid_large.jsonl

# 4. Score with LongMemEval's own scorer (deterministic, ~1 s)
python print_retrieval_metrics.py out/full500_hybrid_large.jsonl
```

The two model-bearing stages, fact extraction and embedding, cost approximately $1.80 and $0.40 respectively, for a total of about $2.20 in OpenAI usage. Wall-clock time from a cold cache is roughly 55 minutes, dominated by the single extraction call per unique session. Both the embeddings and the extracted facts are written to disk, so any later run, including every ablation reported below, reuses them and incurs no further API cost.

3 Results

3.1 Headline comparison

Grove surfaces a gold evidence session within its top five on 99.36% of the 470 questions. Against the principal retrieval-only baselines, this leads gbrain's hybrid configuration (97.60%), gbrain's vector-only configuration (97.40%), and MemPalace's raw baseline (96.60%) by 1.76 to 2.76 points. A separate MemPalace configuration adds a Claude reranker at query time; its reported figures, the relationship of those figures to the published benchmark, and the cost implications of any query-time LLM rerank are addressed in §5.3, §5.4, and §5.5 respectively. At recall@10, Grove reaches 99.79%; the comparators do not publish a recall@10 figure.

Table 2. Session-level retrieval recall@5 on LongMemEval `_s` (n = 470 non-abstention). Comparator scores come from each system's published writeup. Grove's recall@10 (99.79%) is reported in §3.1; no comparator publishes a recall@10 figure. The MemPalace + Claude reranker configuration is omitted from this table because the project reports it in two non-equivalent forms; see §5.3.

Benchmark	Grove	gbrain (hybrid) [2]	gbrain (vector) [2]	MemPalace (raw) [3]
LongMemEval session recall@5	99.36%	97.60%	97.40%	96.60%

3.2 Per-question-type breakdown

An average over all questions can hide uneven behavior, so we also report recall@5 within each of the six question types, alongside gbrain's published per-type numbers[2]. Grove matches or exceeds gbrain's hybrid configuration on every type. The largest gains land on temporal-reasoning (+4.5), single-session-preference (+3.4), and single-session-user (+2.7), the types where a clean fact index has the most lift over flat retrieval. The remaining types (multi-session, single-session-assistant, knowledge-update) reach the ceiling for both systems.

Table 3. Per-type recall@5, Grove versus gbrain (hybrid). Grove matches or exceeds on every type; gains are largest on temporal-reasoning, single-session-preference, and single-session-user, the categories where a distilled fact index has the most lift over flat retrieval.

Question type	n	Grove	gbrain (hybrid)	Δ
multi-session	121	100.0%	100.0%	+0.0
single-session-assistant	56	100.0%	100.0%	+0.0
single-session-preference	30	96.7%	93.3%	+3.4
temporal-reasoning	127	99.2%	94.7%	+4.5
single-session-user	64	98.4%	95.7%	+2.7
knowledge-update	72	100.0%	100.0%	+0.0
Overall	**470**	**99.36%**	**97.60%**	**+1.76**

4 Ablation

A reasonable objection to any high LongMemEval score is that it may reflect nothing more than a strong embedding model, with the surrounding architecture contributing little. To test that objection directly, we strip the pipeline back to a plain dense retriever and rebuild it one component at a time, measuring recall@5 at each step.

Table 4. Pipeline ablation, n = 470. The +0.85 from the model swap is the gain a vector retriever recovers from a stronger embedder alone; the +2.34 from extraction and the hybrid combiner is Grove's pipeline contribution on top of an equally-strong vector retriever.

Configuration	R@5	Δ
Dense, `text-embedding-3-small`, full-session text only	96.17%	baseline
Dense, `text-embedding-3-large@1536`, full-session text only	97.02%	+0.85
+ `gpt-4.1-mini` fact extraction + hybrid max-combiner (Grove)	**99.36%**	**+2.34**

The first step, swapping the smaller embedding model for the larger one used throughout this paper, accounts for +0.85. This is the gain available to any vector retriever simply from a better embedder, and it is the part of the result that does reduce to embedding strength. The second step, adding fact extraction and the hybrid max-combiner on top of that same strong embedder, accounts for a further +2.34. That figure is Grove's pipeline contribution, measured against an equally-strong vector baseline, and it is exactly the quantity the objection above leaves unexplained.

To check that the +2.34 contribution does not depend on the specific small LLM used for extraction, we repeated the full pipeline with `gpt-4o-mini-2024-07-18` (the snapshot used by TiMem[9]) in place of `gpt-4.1-mini`. Recall@5 lands at 98.94%, a difference of 0.42 points (two questions on n = 470). The pipeline's contribution therefore reflects the extract-then-hybridize structure rather than any particular extraction model. We retain `gpt-4.1-mini` as the published configuration because it is the small-model standard used across 2026 memory benchmarks[10][11].

It is worth being explicit about what this number does not include. Grove's headline result is produced without any of the graph, bitemporal, identity, or provenance machinery that the production system runs; none of those layers is active in this evaluation, because LongMemEval's retrieval track does not pose questions that would exercise them. They contribute to the product, not to this score.

5 Discussion

5.1 What LongMemEval measures

LongMemEval scores exactly one capability: given a long chat history, can the system surface the session that holds the evidence for a question? That is a flat retrieval task over session-sized units with a fixed top-k cutoff. Its value is that it makes a memory claim falsifiable, which is rare in this space and the reason we report against it. Its limit is that a single flat-retrieval number is not, on its own, a complete measure of memory-system quality.

5.2 What it does not measure

The retrieval track does not exercise any of the capabilities below. We list them not as a roadmap but to calibrate the headline number, so that it is read as evidence of strong flat retrieval and nothing more.

- **Multi-hop relational queries.** A question such as "Who is blocking the project owned by the person who reports to Bob?" can only be answered by traversing typed relationships between entities. Vector retrieval over raw chat returns text that is superficially similar to the question; it cannot follow a chain of relationships, whereas a typed graph can return the actual path.
- **Bitemporal state.** A question such as "What was true on April 12, before the correction on April 20?" requires tracking, for every fact, both when it became valid and when it was superseded, and then answering as of a chosen point in time. Vector similarity has no notion of valid-from or valid-until and cannot distinguish a current fact from a refuted one.
- **Identity resolution.** Answering "What did the CTO decide?" requires recognizing that "the CTO", "@bob", "BM", and "Bob Martinez" denote the same person and resolving them to a single canonical entity. Fuzzy text matching degrades as soon as the surface forms diverge.
- **Provenance.** A request to "cite the source for X" requires every stored fact to carry a back-pointer to its origin: a message id, a commit hash, a meeting note. Retrieval over chat text can return the passage that surfaced, but it does not certify that the passage actually supports the claim.

5.3 Comparator integrity

gbrain[2] publishes a fully documented methodology: pinned dataset and code commits, the scorer fixed to LongMemEval's upstream version, fixed random seeds, randomized query order, and a sealed set of relevance judgments at the adapter boundary. Both of its reported configurations, hybrid (97.60%) and vector-only (97.40%), use `text-embedding-3-large` at

1536 dimensions, with the hybrid variant adding BM25 lexical scores fused by reciprocal rank fusion. These numbers reproduce from the public repository, which is why we treat them as the reference point for this comparison. A separate gbrain result, *BrainBench*, is measured on a proprietary corpus and has not been independently replicated; we do not engage it here.

MemPalace[3] is a more complicated case. Its reported 96.60% recall@5 has been reproduced by several independent parties[6][7][8], who find that the figure comes from ChromaDB's default `all-MiniLM-L6-v2` embeddings indexing the raw transcript verbatim, with none of MemPalace's namesake "palace" structure engaged. When that structure is turned on, the same third-party reproductions report the score falling to 89.4% and 84.2%. A separate configuration, MemPalace augmented with a Claude (Haiku or Sonnet) reranker at query time, is reported by the project in two non-equivalent forms: 100.0% recall@5 on the full 500-question set after three manual patches against the evaluation set, and 98.4% recall@5 on a 450-question held-out subset without those patches. The held-out 98.4% is the figure most directly defensible against overfitting concerns, and is the number we carry into the architectural-fairness analysis (§5.4) and the cost analysis (§5.5). We do not list the reranker configuration in Table 2 because a single cell cannot honestly represent both reported forms. The 96.60% baseline carried into Table 2 is the headline number MemPalace cites as its own, and we record its provenance here so that the reader can weigh it accordingly.

A precision point on the gbrain comparison: at pure dense retrieval with the same embedding model and dimensionality, gbrain (vector) (97.40%) is approximately 0.38 points above Grove's matched pure-dense baseline (97.02%, line 2 of Table 4). The gap sits within the range of implementation-detail variance — chunking unit, preprocessing, normalization — that two independent runs of "dense retrieval with the same embedder" produce on this benchmark. Grove's published 99.36% comes from the extraction-and-hybrid stage built on top of that baseline, isolated as +2.34 in §4.

5.4 Architectural fairness

The five systems compared in Table 2 are not architecturally equivalent. To make the comparison precise, we split them into two tiers along a single axis: whether the system invokes a large language model at any point in the retrieval pipeline.

Table 5. Comparator architectures grouped by LLM use. Both Grove and MemPalace + reranker introduce an LLM into the pipeline, but at different stages.

Tier	System	Index-time LLM	Query-time LLM	R@5
Retrieval-only	Grove (dense baseline)	none	none	97.02%
Retrieval-only	gbrain (vector)	none	none	97.40%
Retrieval-only	MemPalace (raw)	none	none	96.60%
LLM-augmented	**Grove (published)**	**gpt-4.1-mini, per session**	**none**	**99.36%**
LLM-augmented	MemPalace + Claude reranker (held-out, n=450)	none	Claude Haiku/Sonnet, per query	98.4%
Lexical-augmented	gbrain (hybrid)	none	none (BM25 + RRF)	97.60%

Within the retrieval-only tier, the three systems sit within 0.8 points of each other; differences at this level fall inside the implementation-detail variance discussed in §5.3. Within the LLM-augmented tier, Grove (99.36% on n = 470) and MemPalace + Claude reranker (98.4% on its 450-question held-out subset, see §5.3) both clear 98%, and the substantive distinction is where the language model is invoked rather than the score gap. gbrain (hybrid) (97.60%), which augments retrieval with BM25 lexical scores fused by reciprocal rank fusion rather than with a language model, sits below both.

The two LLM-augmented systems differ in *where* the language model is invoked. Grove invokes it once per session, at index time, to produce a static fact index that subsequent queries read from. MemPalace + reranker invokes it once per query, at retrieval time, to rescore candidate sessions returned by the underlying vector store. The accuracy outcome is comparable; the cost implications are not, and we quantify them in §5.5.

5.5 Cost analysis

The cost class implied by a query-time LLM reranker differs materially from the cost class implied by index-time fact extraction. We quantify this contrast for Grove and the MemPalace + Claude reranker configuration discussed in §5.3, the two systems in this comparison that introduce a language model into the retrieval pipeline. The contrast is structural; it does not depend on which of the reranker's reported accuracy figures is taken as canonical.

Grove pays for fact extraction once, when a session is indexed. With LongMemEval _s's 19,195 unique sessions and OpenAI's `gpt-4.1-mini` pricing at the time of writing, this totals approximately $5. Every subsequent query incurs a single embedding API call costing approximately 6×10^{-6} (about six millionths of a dollar); the cosine similarity that follows is computed in-memory at no API cost.

MemPalace + Claude reranker pays nothing at index time but issues a Claude API call on every user query. The reranker call processes the query alongside roughly 30 candidate sessions, totaling on the order of 15,000 input tokens at Anthropic's Claude Haiku 4.5 pricing[12], costing approximately $0.015 per query.

Table 6. Cumulative API spend for retrieval on a stable corpus, at increasing query volume. Index-time cost is amortized once; query-time cost scales linearly. Figures are approximate and drawn from 2026 published pricing for the named models.

Queries served	Grove (this paper)	MemPalace + Claude reranker	Ratio
1	~$5	~$0.02	Grove front-loads
~330	~$5	~$5	break-even
10,000	~$5	~$150	30× cheaper
100,000	~$6	~$1,500	250× cheaper
1,000,000	**~$15**	**~$15,000**	**1,000× cheaper**

The cost contrast is the practical distinction between paying once at index time and paying on every retrieval. For any deployment with non-trivial query volume, the choice has material consequences regardless of the specific accuracy figure cited for the reranker variant. We report the cost contrast independent of any direct accuracy claim against MemPalace + reranker; its reported figures and their provenance are addressed in §5.3.

6 Limitations

Several boundaries on the result above are worth stating directly.

- We do not run the end-to-end QA-accuracy track. Retrieval quality and answer quality are distinct: a system can surface the right session and still answer poorly under a given reader, and the reverse is also possible. This paper speaks only to the first.
- The result is on the `_s` split, whose histories run to roughly 115,000 tokens. The larger `_m` split, with histories near 1.5 million tokens, is the harder variant and is not evaluated here.
- We do not benchmark end-to-end cost or latency. The pipeline's wall time is dominated by one extraction call per unique session, which is acceptable for a batch evaluation; the production system indexes incrementally as sessions arrive rather than extracting the whole corpus at once, and that path is not what LongMemEval measures.
- The extraction prompt was written once and never tuned against the LongMemEval data. No held-out validation set was used to adjust it, so the result reflects an untuned prompt rather than a best case.

References

[1] Wu, D. et al. *LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory.* ICLR 2025. arXiv:2410.10813

[2] Tan, G. *gbrain-evals.* 2026. github.com/garrytan/gbrain-evals, in particular `docs/benchmarks/2026-05-07-longmemeval-s.md`.

[3] MemPalace project. *Benchmarks.* github.com/MemPalace/mempalace/blob/develop/benchmarks/BENCHMARKS.md

[4] LongMemEval cleaned dataset on Hugging Face. huggingface.co/datasets/xiaowu0162/longmemeval-cleaned

[5] LongMemEval source code. github.com/xiaowu0162/LongMemEval, scorer at `src/retrieval/eval_utils.py` and `src/evaluation/print_retrieval_metrics.py`.

[6] Vectorize. *MemPalace Benchmarks Review.* vectorize.io/articles/mempalace-benchmarks

[7] Coding With Cody. *MemPalace: Digital Castles on Sand.* April 2026. codingwithcody.com/2026/04/13

[8] MemPalace GitHub Issue 875, misrepresentation of benchmark methodology. github.com/MemPalace/mempalace/issues/875

[9] Liu, Y. et al. *TiMem: Time-aware Memory for AI Agents.* arXiv 2601.02845. arxiv.org/abs/2601.02845 — uses `gpt-4o-mini-2024-07-18` for memory consolidation; cited as the reference point for our extraction-model robustness check in §4.

[10] Prosus AI Tech Blog. *MemEval: Benchmarking Memory for AI Agents.* March 2026. medium.com/prosus-ai-tech-blog/memeval-benchmarking-memory-for-ai-agents — standardizes `gpt-4.1-mini` across nine systems.

[11] MemMachine project. *MemMachine: Memory Architecture for Long-running AI Agents.* arXiv 2604.04853. arxiv.org/abs/2604.04853

[12] Anthropic. *Claude API pricing reference.* anthropic.com/pricing — Claude Haiku 4.5 used as the per-query reranker pricing basis in §5.5.

Appendix A · Extraction prompt

The system prompt below is issued to `gpt-4.1-mini` for fact extraction, with the output format constrained to a JSON object:

```
You distill a chat session into atomic, retrievable facts about the user.
Output a JSON object {"facts": [...]} where each fact is one short,
self-contained sentence in third person about the user: preferences,
decisions, attributes, plans, or events they reported. Each fact should
be standalone (no pronouns referring outside). If the session has no
user-stated facts, return {"facts": []}. Keep facts terse.
```

The accompanying user message carries the full session transcript with role labels, truncated at 32,000 characters, which is well inside `gpt-4.1-mini`'s context limit. The temperature is fixed at 0 so that extraction is deterministic. Run across the full `_s` haystack of 19,195 unique sessions, the extractor produced 111,969 facts in total; sessions made up of generic filler material (the ShareGPT and UltraChat conversations used as distractors) were correctly recognized as carrying no user facts and returned empty arrays.

Appendix B · Pipeline output schema

The scorer reads one JSONL row per scored question. Each row carries the question's metadata, the rank-ordered list of retrieved session ids, and the per-question metric values:

```
{
  "question_id":   "<LongMemEval question id>",
  "question_type": "<one of the six LongMemEval types>",
  "retrieval_results": {
    "metrics": {
      "session": {
        "recall_any@5":  <float in [0,1]>,
        "recall_all@5":  <float in [0,1]>,
        "ndcg_any@5":    <float in [0,1]>,
        "recall_any@10": <float in [0,1]>,
        "recall_all@10": <float in [0,1]>,
        "ndcg_any@10":   <float in [0,1]>
      },
      "turn": { }
    },
    "top10_sessions": [ "<session id>", ... ]
  }
}
```

LongMemEval's `print_retrieval_metrics.py` reads rows in this shape, drops the `_abs` abstention questions, and averages each metric over the questions that remain. Feeding our pipeline's output to that script unmodified reproduces the 99.36% headline.