COMET: A FRAMEWORK FOR DIAGNOSIS-GUIDED RETRIEVAL ADAPTATION IN AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent memory systems increasingly refine how experience is stored, but memory retrieval often remains reliant on general-purpose dense embedders. We introduce COMET, a framework for connecting retrieval-failure diagnosis with targeted adaptation through five recurring failure patterns: **c**oncentration, **o**ver-fragmentation, **m**emory-query form mismatch, **e**ntity mismatch and **t**emporal mismatch.

First, we **diagnose** which failure patterns are most prominent in long-term memory retrieval. Across three memory systems and two benchmarks (LoCoMo and PersonaMem), we operationalize the five failure patterns as measurable proxies, finding that they significantly predict held-out retrieval failures beyond standard controls. We find particularly substantial evidence for over-fragmentation, memory-query form mismatch and entity mismatch. Building on these results, we **adapt** three embedding models across families and scales by constructing 1250 queries from an independent dataset, with 250 targeting each of the five failure patterns.

We find that O-250, M-250 and E-250, the three datasets corresponding to the most prominent failure patterns, generally produce the most effective LoRA adapters. OME-750, the adapter trained from their union, is our best-performing adapter overall, generally outperforming all 1250 queries combined; on Qwen3-Embedding-0.6B, it improves mean Recall@10 by **7.70 percentage points** above base and by **3.08 points** above matched-compute training on untargeted memory-retrieval datasets. Final answer generation accuracy also improves on LoCoMo by **8.57 points**. The adapter further improves nDCG@10 on eight benchmarks from the LMEB suite by **8.22 points**, matching EvoEmbedding-0.8B, a memory-specialized embedder trained with **245 times** as many independent queries. These findings suggest a general principle connecting memory-retrieval failures and retrieval adaptation: first diagnose, then adapt.

1 INTRODUCTION

Long-term memory systems allow agents to retain user facts, preferences, and events across conversations (Park et al., 2023; Packer et al., 2023), but their usefulness depends on whether the correct evidence can be retrieved when new questions arrive. The most prominent long-term memory systems use RAG, i.e. retrieval-augmented generation (Lewis et al., 2020), as a foundation: the system collects a **memory store** from conversations and retrieves stored memories through dense-embedding-based retrieval. Modern memory systems invest heavily in varied storage architectures: Mem0 summarizes conversations into stored atomic facts, A-MEM agentically links between memories, and SimpleMem enriches memories with structured information such as dates and entities (Chhikara et al., 2025; Xu et al., 2025; Liu et al., 2026). Benchmarks such as LoCoMo, PersonaMem, and LongMemEval assess whether agents can recover and use this accumulated experience (Maharana et al., 2024; Jiang et al., 2025; Wu et al., 2024).

Yet a system can store the answer and still fail to retrieve it: retrieval architectures account for about 20 percentage points (pp) of variance in LoCoMo answering accuracy versus about 6pp for storage architectures (Yuan et al., 2026), and embedder performance on general QA benchmarks does not correlate significantly with memory-retrieval performance (Zhao et al., 2026). General-purpose retrievers may miss a person referred to by a description, a preference implicit in past behavior, or evidence spread across several memories; despite this, most memory systems have opted to

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.

employ such general embedding models (Chhikara et al., 2025; Rasmussen et al., 2025). This choice potentially forgoes large accuracy gains obtained through embedding models adapted to memory retrieval, which have only recently been explored in EvoEmbedding (Nie et al., 2026) and HiNS (Tian et al., 2026).

In this paper, we outline a framework for efficient retrieval adaptation of embedding models to long-term memory. Our starting observation is that RAG-based memory storage architectures exhibit five categories of recurring failure patterns that can be identified and diagnosed: **c**oncentration, **o**ver-fragmentation, **m**emory-query form mismatch, **e**ntity mismatch, and **t**emporal mismatch (**COMET**).

Our contributions are as follows:

- We propose an interpretable framework that comprises five recurring patterns of retrieval difficulty in long-term memory tasks, diagnoses which of the patterns are most prevalent, and supports efficient memory-retrieval adaptation through targeting the diagnosed patterns.
- We provide **empirical validation** of these five patterns through regressions and robustness tests across three memory systems and two benchmarks, finding that three of the five are especially prominent (over-fragmentation, memory-query form mismatch and entity mismatch).
- We demonstrate **diagnosis-guided adaptation**: small datasets targeting the diagnosed COMET categories outperform matched unaugmented and generic-rewrite controls on aggregate retrieval metrics across three backbones, with gains extending to answering and external retrieval.

2 Related work

Diagnosing memory retrieval failures. The closest prior work is Leung et al. (2026), whose fine-grained taxonomy of RAG errors uses binary yes-or-no judgments without statistical validation and does not foreground failure patterns especially salient in conversational memory, such as entity and temporal mismatch. COMET places these patterns at the forefront of its design and uses continuous proxies validated by regression on held-out retrieval failure. Other diagnoses of end-to-end memory-system failures (Yuan et al., 2026) generally employ the coarse categories of write, retrieval and utilization failure instead of fine-grained descriptive categories. Moreover, COMET uses the diagnosed failure patterns to guide what training data to construct for adaptation instead of remaining diagnosis-only.

Memory storage and query interventions. Past retrieve-then-generate memory systems (Lewis et al., 2020) have proposed various enrichments to a raw-turn RAG baseline that target specific patterns of failures. **Mem0** (Chhikara et al., 2025) compresses turns to atomic facts; **SimpleMem** multi-indexes memories through entities and timestamps (Liu et al., 2026); **A-MEM** continually evolves a memory's stored representation by linking it to related memories (Xu et al., 2025); and **QueryLink** (Hu et al., 2026) proposes query-side intervention through expanding a query's entities, keywords and implicit information.

Specialized embedding models for memory retrieval. The vast majority of memory systems utilize dense-embedding retrieval (Lewis et al., 2020; Chhikara et al., 2025; Liu et al., 2026). Despite research showing that embedding models suited for general QA datasets may not transfer well to memory retrieval (Zhao et al., 2026), few papers provide a precedent for memory-specific embedders. PerLTQA (Du et al., 2024) synthesizes a memory-retrieval dataset but predates long-term memory benchmarks like LoCoMo and PersonaMem; HiNS and EvoEmbedding fine-tune a modern embedding model evaluated on modern memory systems and downstream tasks, and EvoEmbedding (Nie et al., 2026) additionally changes the encoder architecture to maintain evolving representations. COMET uses an unmodified encoder and differs from both routes in training data construction, selecting it from the diagnosed failure profile rather than from generic question synthesis or architectural change.

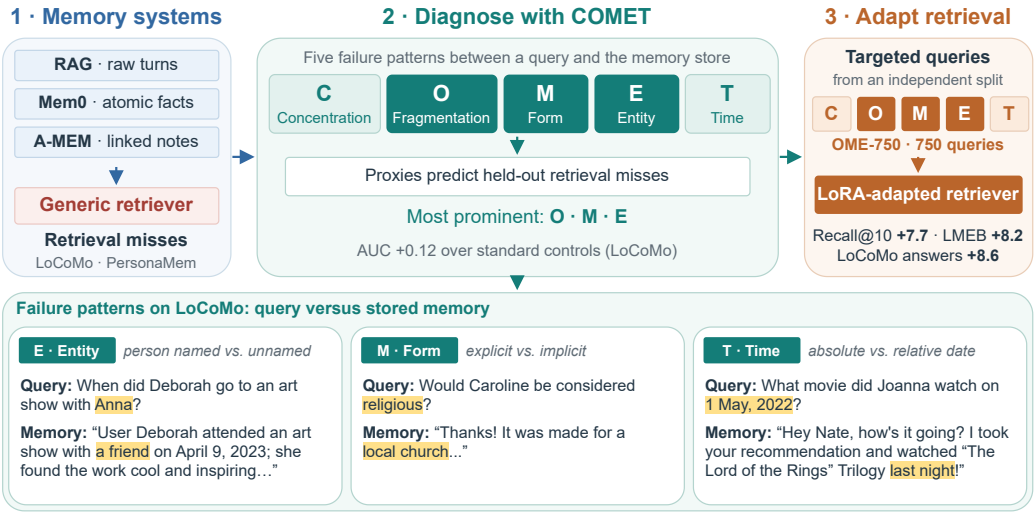

Figure 1: COMET connects retrieval-failure diagnosis to targeted adaptation: proxies diagnose the most prominent failure patterns (O, M, E), which guide training-data construction on an independent split. Bottom: LoCoMo examples, mismatched spans highlighted.

3 COMET: FROM DIAGNOSIS TO ADAPTATION

3.1 PROBLEM FORMULATION

We study an agent that writes conversational history into a memory store, retrieves the top-k memories for a query, and passes them to an answerer. Our focus is the retrieval stage: the required evidence exists in the store, but the retriever may not recover it. We freeze each store throughout diagnosis and adaptation evaluation, separating changes to retrieval from changes to stored information.

For each query, we annotate the following using an LLM. First, its ground-truth memory set, which identifies the minimal sufficient set of memories needed to answer the query correctly. This set supports our downstream retrieval metrics, where we measure the fraction recovered (Recall@k) and whether the complete set is recovered (binary retrieval failure). Second, we annotate the entities (people, places, events etc.) and time expressions within the queries and memories, as well as the portions of the memory text judged to be relevant to answering the query. A more detailed mathematical formulation of the problem, as well as the annotation pipeline and eligibility rules, are found in Appendix A.

3.2 THE COMET FAILURE PATTERNS

COMET describes five non-exclusive failure patterns in memory retrieval based on relationships between a query and its memory store (Figure 1). Evidence can be **c**oncentrated too thinly within a memory (C), **o**ver-fragmented across memories (O), or expressed differently in the **m**emory versus the query in implicit forms (M), **e**ntities (E), or **t**ime references (T). Similar-looking distractors can further obscure the relevant evidence. These patterns provide a vocabulary for both describing failures and designing examples that expose a retriever to the corresponding difficulty; to quantify them, we compute continuous proxies based on the annotations above, orienting each so that higher values indicate more adverse retrieval conditions (Table 1). Appendix A gives the precise definitions of all proxies and Appendix F the annotation prompts.

Over-fragmentation is simply proxied by the number of ground-truth memories required to answer a query; though a higher number of ground-truth memories is mathematically expected to make complete retrieval more difficult, it has no expected mathematical relationship to Recall@k. For this reason, we include regressions with Recall@10 as the dependent variable in Appendix C. For a discussion on whether the proxies capture distinct failure patterns that meaningfully differ from general lexical overlap, see Experiments.

Table 1: The five COMET failure modes and the proxy variables used to quantify them. For paired proxies, $^+$ compares the query with its ground-truth memories and $^-$ with non-ground-truth (distractor) memories.

COMET category	Intended failure pattern	Proxies	Proxy meaning
Concentration	Only a small portion of the ground-truth memory is relevant to the query	$C(q)$	One minus the fraction of ground-truth memory text relevant to the query
Over-fragmentation	Relevant information is fragmented across multiple ground-truth memories	$O(q)$	Number of separately retrievable values in the minimal answer-sufficient set
Memory-query form mismatch	The query is only implicitly answered by the memory	$M^+(q), M^-(q)$	Document-frequency-weighted lemma overlap with the query
Entity mismatch	Different expressions refer to the same entity	$E^+(q), E^-(q)$	Similarity between the entities in the query and in the memory
Temporal mismatch	Different expressions refer to the same time	$T^+(q), T^-(q)$	Similarity between the time expressions in the query and in the memory

3.3 AN OVERVIEW OF DIAGNOSIS-GUIDED ADAPTATION

Diagnosing and validating failure patterns. Using the eight continuous proxies described above, alongside a set of standard control variables described in Appendix C, we fit Firth bias-reduced logistic regressions (Firth, 1993) with binary retrieval failure (complete recall) as the dependent variable. We test whether these proxies predict failures beyond controls, using LoCoMo as our primary benchmark representing long-term memory tasks; we further validate these proxies on PersonaMem-32K.

In total, we fit 15 regressions for LoCoMo and PersonaMem each, one for each benchmark/retrieval configuration/embedding model at $k = 10$ (see Experiments). We compare controls-only to controls-plus-COMET regression models on held-out McFadden-R^2 (pseudo-R^2) and AUC, keeping one cluster (conversation or persona) out of the test set. Following these regressions, we rank the COMET categories by the number of LoCoMo strata it is statistically significant on, using the category's lowest p-value proxy if multiple exist. We also rank the marginal contribution of each category to held-out goodness-of-fit metrics over controls-only, using the highest-contribution proxy if multiple exist. We do **not** expect all COMET categories or proxies to be equally significant; instead, we expect a clear separation where some failure patterns are more well-evidenced than others for selective diagnosis-guided adaptation.

We diagnose on LoCoMo, which targets long-term memory, whereas PersonaMem targets user profiling and personalized responses, and LoCoMo has 2.6 times as many questions (1,534 versus 589); Appendix C reports individual strata, $k = 20, 50, 100$, and Recall@10 as the dependent variable.

Turning patterns into training data. For each COMET category, a construction agent augments queries from PerLTQA, a previous memory-retrieval dataset, to expose one COMET failure pattern while preserving complete, unmodified memory texts; Appendix Table A1 lists a description of the transformations. The construction agent first selects eligible questions for these augmentations, filtering for structural properties such as the presence of entities and time expressions, and then rewrites the queries according to the failure patterns. No augmentation changes the source memories; as nearly all PerLTQA queries are single-memory (31 of 5,266 have more than one), the over-fragmentation dataset necessarily changes which memories are ground-truth for a query.

To isolate the benefit provided by COMET targeting versus generic memory-retrieval adaptation, we compare to several control datasets: 1) PerLTQA without selection or augmentation; 2) PerLTQA using the selected questions but without augmentation; 3) PerLTQA using generic, non-COMET-targeted augmentation; 4) the hardest PerLTQA questions for each base embedder, without augmen-

tation (Generic-Hard-250). We hypothesize that adaptation can be **diagnosis-guided**: the COMET failure patterns diagnosed as most significant on LoCoMo and our tested memory systems will provide the most effective adapters.

Adapting retrievers for contrastive learning. Using PerLTQA's official train split (5,266 queries), we construct 250 augmented queries per COMET category to produce specialist datasets, C-250, O-250, M-250, E-250 and T-250. COMET-1250 trains a single adapter on their concatenation; as all queries are augmented, no two queries overlap. Training for LoRA adapters (Hu et al., 2021) is conducted using a standard contrastive objective (softplus loss) to increase similarity to labelled ground-truth memories versus mined hard negatives, via a fixed two-epoch schedule, to make same-sized datasets comparable in number of training steps; all adapters are trained across several base embedding models and random seeds, detailed in Experiments. The objective and complete configuration are in Appendix B.

4 EXPERIMENTS

We organize the evaluation around two questions. **RQ1**: do structural failure patterns explain memory retrieval behavior? **RQ2**: does data construction guided by diagnosed failure patterns improve adaptation?

Memory systems, embedders, and retrieval budgets. We evaluate RAG (one dated, speaker-labeled turn per memory), Mem0 (distilled atomic facts), and A-MEM (turns enriched with summaries, concepts, and links). The main benchmarks are LoCoMo and PersonaMem-32K. Our primary embedding model backbone is Qwen3-Embedding-0.6B (Zhang et al., 2025); Qwen3-Embedding-4B tests scale transfer and BGE-large-en-v1.5 tests cross-family transfer (Xiao et al., 2023).

We use retrieval budget $k = 10$, which avoids benchmark saturation and matches EvoEmbedding's LMEB retrieval budget ($k = 10$). The primary diagnosis results use native retrieval for Mem0 and A-MEM: both use dense-embedding retrieval, but Mem0 natively employs a lexical-fusion with BM25 and an entity-matching boost, and A-MEM uses graph traversal to linked nodes. Adapter comparisons use dense-only retrieval at $k = 10$, isolating embedding changes from lexical fusion or graph traversal. We report the unweighted mean over the six benchmark-system cells, averaged across three training seeds (42, 43, 44); complete per-cell results appear in Appendix D.

Answer generation and external transfer. We additionally evaluate the primary adapters to assess whether retrieval gains translate to gains in answer generation accuracy. A fixed answering model (gpt-5.6-luna) receives the retrieved memories under $k = 10$; LoCoMo assesses accuracy through an LLM correctness judge plus deterministic F1-score, and PersonaMem uses multiple-choice option exact match.

To compare with the published EvoEmbedding results, we employ eight benchmarks used by EvoEmbedding within the external Long-horizon Memory Embedding Benchmark (LMEB) suite: QASPER, PeerQA, CovidQA, MLDR, ESG-Reports, REALTALK, LoCoMo, and LongMemEval (Zhao et al., 2026). All except LoCoMo are held-out datasets for this paper. Following the official protocol, we report the adapters' nDCG@10 averaged across all eight tasks.

4.1 RQ1: DO STRUCTURAL FAILURE PATTERNS EXPLAIN MEMORY RETRIEVAL BEHAVIOR?

COMET adds predictive information beyond standard controls. Across all 15 LoCoMo strata, the proxies improve held-out AUC by 0.116, with pseudo-R^2 gains of 0.135 (Table 2); on the 15 PersonaMem strata, AUC improves by 0.145 and pseudo-R^2 by 0.156. These are held-out predictions across conversations or personas, not just improvements in training fit. Full numerical tables and Recall@10 regressions are in Appendix C.

Notably, **over-fragmentation is not a mathematical artifact of binary retrieval failure**: it is significant on 15 out of 15 strata with Recall@10 as the dependent variable on LoCoMo as well. Appendix A elaborates on the lack of an expected mathematical relationship between the over-fragmentation proxy and Recall@10; Appendix C presents detailed results. Neither do the proxies describe general lexical similarity, or overlapping information. Inter-category correlations between proxies are only modest ($|\rho| < 0.3$), all proxies are correlated at $|\rho| < 0.4$ with general lexical overlap, and including it as a control does not remove their significance; as Table C3 describes

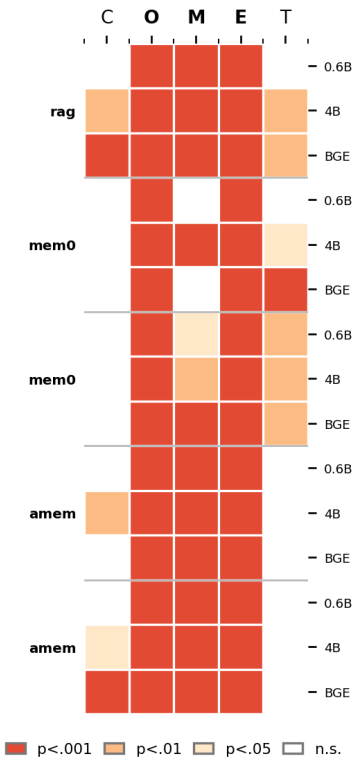

Table 2: Diagnosis evidence by COMET category. For each COMET category (C = concentration, O = over-fragmentation, M = memory-query form mismatch, E = entity mismatch, T = temporal mismatch): the number of LoCoMo strata (of 15) on which it is significant. The All rows include all COMET proxies and compare controls-only and controls-plus-COMET held-out fit, averaged over all benchmark strata (gain in parentheses). Bold marks the three most substantially evidenced categories.

COMET category	Significant strata (15 total)	Held-out AUC gain	Held-out pseudo-R^2 gain
C	5	0.000	0.000
O	**15**	**+0.042**	**+0.054**
M	**13**	**+0.022**	**+0.014**
E	**15**	**+0.058**	**+0.048**
T	7	+0.009	+0.004
All, LoCoMo	-	0.703 → 0.819 (+0.116)	0.116 → 0.251 (+0.135)
All, PersonaMem	-	0.656 → 0.801 (+0.145)	0.054 → 0.210 (+0.156)

Figure 2: Category significance across the 15 LoCoMo $k = 10$ strata. Each row is one stratum, the five retrieval configurations crossed with the three evaluation embedders; each column is one COMET category shown by its most-significant proxy. Color indicates significance level of the proxy.

(Appendix C), 12 of the 13 significant category-level associations on the pooled LoCoMo strata remain significant.

We also note **between-system and between-benchmark failure profile variation**: entity and form mismatch recur across all three systems, over-fragmentation is strong throughout LoCoMo but significant only for Mem0 on PersonaMem, and concentration is stronger on PersonaMem than on LoCoMo. Mem0 generally shows the strongest fragmentation results and weakest concentration results, whereas A-MEM shows the strongest concentration and also contains the longest memories. This variation may be driven by both inter-benchmark differences, where only 13.9% of PersonaMem questions are multi-memory versus 26.8% on LoCoMo, as well as memory-system mechanisms, where Mem0's atomic fact compression concentrates relevant information at the cost of fragmentation.

4.2 RQ2: Does data construction guided by diagnosed failure patterns improve adaptation?

RQ1 diagnoses over-fragmentation, memory-query form mismatch and entity mismatch as the most significant failure patterns, with a large drop-off to temporal mismatch. We therefore also train OME-750 on the union of O-250, M-250 and E-250, and test whether these adapters outperform both their matched controls and the C-250, T-250 and COMET-1250 adapters.

OME-750 surpasses all controls on all three backbones, with the strongest mean primary retrieval result on Qwen3-0.6B and BGE-large and 0.1pp below COMET-1250 on Qwen3-4B (Table 3). On Qwen3-0.6B it improves Recall@10 by 7.70 points over base, 3.08 over PerLTQA-750 (750 unaugmented PerLTQA queries), and 1.75 over Placebo-750 (750 generically-augmented PerLTQA queries), averaged over PersonaMem and LoCoMo. Its gains over base are 13.17 points on Qwen3-4B and 4.34 on BGE-large. The six-cell macro improvements therefore replicate across

Table 3: Retrieval adaptation, averaged over the six benchmark–system cells. Gains in Recall@10 over each backbone's base, averaged over the three memory systems, three training seeds, and both benchmarks (LoCoMo and PersonaMem); the base row gives absolute Recall@10. Bold marks COMET adapter results that outperform at least one matched control, supported by paired bootstrap tests in at least two of three seeds (Appendix Table D7). Per-benchmark results are in Appendix D.

Training data	Queries	Qwen3-0.6B	Qwen3-4B	BGE-large
Base	0	63.06	58.77	66.42
PerLTQA-250	250	+5.41	+9.37	+1.13
Placebo-250	250	+5.13	+10.31	+1.57
Generic-Hard-250	250	+4.11	+7.03	−1.58
C-250	250	+4.90	+6.90	**+2.18**
O-250	250	**+6.10**	+9.05	**+3.11**
M-250	250	**+7.19**	**+10.42**	**+3.15**
E-250	250	**+6.37**	**+10.91**	**+2.82**
T-250	250	+4.74	+9.27	+1.60
PerLTQA-750	750	+4.62	+11.59	+2.45
Placebo-750	750	+5.95	+11.65	+2.22
OME-750	750	**+7.70**	**+13.17**	**+4.34**
PerLTQA-1250	1,250	+4.35	+12.41	+2.84
Placebo-1250	1,250	+5.31	+11.86	+3.56
COMET-1250	1,250	**+6.78**	**+13.27**	**+3.95**
PerLTQA-Full	5,266	+2.59	+9.80	+3.64

model family and scale, without implying improvement in every individual cell. All comparisons to controls are substantiated with paired bootstrap tests using 2000 replications in Appendix D.

Diagnosis-guided adaptation is effective as a framework on the evaluated benchmarks and systems. The three failure patterns diagnosed most strongly in RQ1 generally produce the strongest adapters (O-250, M-250, E-250); these adapters significantly outperform at least one matched control on 8 out of 9 cells above, while the two remaining adapters together (C-250, T-250) do so in only one. These results do not suggest causality but do indicate that effective adaptation can be driven by targeting empirically diagnosed failure patterns.

Retrieval gains transfer to answering; OME-750 raises LoCoMo judged accuracy by 8.57 points. PersonaMem accuracy moves under 1pp for every retriever, as it depends much less on retrieval (20–25pp accuracy gap between full and zero recall, against 62–64pp on LoCoMo; Appendix D). Prior literature such as EvoEmbedding likewise finds PersonaMem nearly flat across embedders, attributed to the multiple-choice format. In its own pipeline and with a different answering model, EvoEmbedding-0.8B loses 7.15pp on LoCoMo against Qwen3-Embedding-0.6B under an LLM judge at $k = 8$, the k-value closest to ours. Averaged across $k = 1$, 2, 4, 8 and 16, it changes LoCoMo and PersonaMem by -1.77 and -0.24 over Qwen3-Embedding-0.6B, and its largest LoCoMo gain at any k is +6.56 ($k = 2$), below our +8.57, indicating that our margin is not a k-driven artifact. EvoEmbedding-4B, however, gains a strong 14.08 on LoCoMo over Qwen3-Embedding-4B (Nie et al., 2026).

External transfer with few training queries. On the eight-benchmark LMEB suite, OME-750 raises Qwen3-Embedding-0.6B nDCG@10 from 50.87 to 59.09. Its performance is comparable to EvoEmbedding-0.8B (58.97), which uses 184,137 training queries against our 750 (Table 5). In particular, OME-750 outperforms EvoEmbedding on all three long-term memory benchmarks (LoCoMo, LongMemEval, REALTALK) and two conversational-QA or document retrieval tasks (MLDR, CovidQA), totalling 5 out of 8. At 4B (Appendix Tables D5–D6 give 4B and BGE-large), OME-750 averages below EvoEmbedding-4B but still outperforms it on the same datasets except LoCoMo and MLDR. The data efficiency lies in the number of independent queries (245 times fewer) and training cost: one RTX PRO 6000 (96GB Blackwell) GPU, against EvoEmbedding's reported 26.6 wall-clock hours on eight H800 GPUs (about 213 GPU-hours).

Table 4: End-to-end answering accuracy. Base row absolute; other rows are gains over base (pp), averaged over three systems and seeds; macros average the six cells with LoCoMo judged or F1-scored (deterministic). Bold: significantly above a matched control in at least two of three seeds (Appendix Table D12). EvoEmbedding-0.8B: its reported gain over Qwen3-Embedding-0.6B at $k = 8$, with a different answering model, judge and memory system; not directly comparable.

Training data	Queries	LoCoMo (judge)	LoCoMo (F1)	PersonaMem	Macro (judge)	Macro (deterministic)
Base	0	55.48	46.59	65.93	60.70	56.26
PerLTQA-750	750	+7.29	+4.42	+0.62	+3.96	+2.53
Placebo-750	750	+8.07	+5.12	+0.38	+4.22	+2.75
OME-750	750	**+8.57**	+5.51	+0.36	+4.47	+2.94
COMET-1250	1,250	+7.62	+4.86	+0.62	+4.12	+2.75
EvoEmbedding-0.8B	184,137	−7.15	-	−1.02	-	-

Table 5: External transfer on the eight-task LMEB suite, Qwen3-Embedding-0.6B. Absolute nDCG@10, averaged over the eight benchmarks and three training seeds, with the number of training queries and the training cost of each model. Bold marks COMET adapter results that outperform at least one matched control, supported by paired bootstrap tests in at least two of three seeds (Appendix Table D8); OME-750 outperforms both matched 750-query controls. EvoEmbedding-0.8B is as reported by Nie et al. (2026) under the same protocol (one model per scale).

Model	Queries	nDCG@10	GPU-hours
Qwen3-Embedding-0.6B (base)	0	50.87	0
PerLTQA-750	750	57.49	~1
Placebo-750	750	57.15	~1
OME-750 (Qwen3-Embedding-0.6B)	750	**59.09**	**~1**
PerLTQA-1250	1,250	57.20	~2
Placebo-1250	1,250	57.23	~2
COMET-1250 (Qwen3-Embedding-0.6B)	1,250	57.67	~2
EvoEmbedding-0.8B	184,137	58.97	~213

Do the examples instantiate the intended difficulties? Paired measurements before and after rewriting show that the C, O, E, and T datasets increase their targeted proxy measures (Appendix Table D13). Memory-query form mismatch (M-250) is the exception: its lexical proxy changes little (0.788 to 0.779), with a source median already at 1. This may indicate that the memory-query form proxy can distinguish between cases of low mismatch and high mismatch (explicit overlap between query and memory versus little to no overlap) but not between different cases of lower overlap. We regard this as an acceptable trade-off for diagnostic evidence, but a genuine proxy limitation.

Random examples (Appendix Tables D14–D19) show the intended rewrites, with M-250 making the query-memory relationship implicit rather than explicit. The C-250, O-250 and E-250 adapters also gain more on LoCoMo queries above the median of their own targeted proxy, i.e. queries exhibiting more of the targeted pattern (Appendix Table D20); this is exploratory, not causal.

Human audits of results. Human audits find generally high validity for time, entity and relevant-span annotations, with the lowest human validity rate on query entities (80%); an independent LLM re-review reproduces the answer judge's labels on all 90 audited cases. Human agreement with the LoCoMo answer judge is 88.9% on a balanced sample and 91.6% reweighted (Appendix E).

Selecting categories on LoCoMo does not drive the result. Diagnosing on an evaluation benchmark raises a circularity or test-set leakage concern, but the diagnosis never uses adapter results, and OME-750's margins over PerLTQA-750 and Placebo-750 are largest on PersonaMem, which was not used for selection: 2.2–3.9pp Recall@10 on every backbone (Appendix Tables D1–D3).

What the controls establish. Each COMET adapter's matched controls have equal numbers of training steps. PerLTQA controls (no selection or augmentation) measure the joint gain from COMET-targeted selection and augmentation; Placebo controls (generic augmentation) isolate COMET-

targeted versus generic augmentation; selection controls, trained on the construction agent's selected source queries without augmentation (Qwen3-Embedding-0.6B, three seeds), are significantly weaker (2.3–3.2pp lower) than O-250, M-250, E-250 and OME-750 (Appendix Table D9); and Generic-Hard-250, trained on the hardest unaugmented PerLTQA questions for each base embedder, falls below PerLTQA-250 on all three backbones, with O-250, M-250 and E-250 significantly above it in every seed on every backbone (Appendix Table D7) and above it on LMEB in eight of nine comparisons (Appendix Table D8), so the COMET gains do not stem from query difficulty alone. COMET-1250, itself matched by PerLTQA-1250 and Placebo-1250, isolates the effect of pruning to the three most strongly diagnosed categories in OME-750. Overall, COMET substantially amplifies adaptation: at equal data, OME-750's gain over base exceeds PerLTQA-750's by an average of **52%** across backbones and Placebo-750's by **46%**, every margin at least 1.5pp. OME-750 outperforms or matches COMET-1250 on all backbones with less data, suggesting that targeting strongly validated failure patterns is more effective than untargeted adaptation; COMET-1250 in turn outperforms at least one of its 1250-query controls on all three backbones (Appendix Table D7), and where it falls below them, on LMEB with Qwen3-Embedding-4B, OME-750 recovers a significant margin over COMET-1250 (Appendix Table D8).

5 Discussion and limitations

A diagnosis-guided framework for memory retrieval adaptation. COMET connects observable retrieval failures to interpretable failure patterns which guide targeted adaptation for memory retrieval systems. Its specific categories make aggregate failures more actionable: a miss associated with a referring-expression mismatch for entities suggests a different training example from a miss involving fragmented evidence. The experiments support the utility of a diagnose-then-adapt principle: targeting empirically validated failure patterns is more effective than generic in-domain or untargeted augmentation, with transfer results in both long-term memory and conversational-QA benchmarks. Here, we instantiate and evaluate that principle only for retrieval adaptation (Appendix A.7 sketches untested storage- and query-side modules for the same patterns); extending it to writing, tool use, or planning would require representations and interventions appropriate to those components. Even within memory retrieval, the current curriculum is shared across all memory systems; though a system-agnostic adapter may be more widely applicable, future research could explore whether system-specific embedding models may provide greater benefits.

What the evidence does and does not establish. The connection between diagnosed patterns and adaptation improvement is strongest at the framework level, but a one-to-one causal mapping between a pattern and a specialist is not established: some failure patterns may overlap, and the lexical proxy for memory-query form mismatch is imperfect and captures only broad mismatches in expression. Annotation uncertainty also affects the measured profiles. The proxies are therefore a tractable operationalization of the failure patterns, not exhaustive or uniquely correct measurements.

The evaluated patterns do not cover all retrieval difficulties or analyze memory storage and utilization (answering) in comparable detail. Retrieval-eligible Mem0 scores exclude annotated write failures (13.05% of LoCoMo and 25.13% of PersonaMem questions; Appendix Table C14), so recall alone does not measure the entire system. Moreover, we do not yet establish scaling potential for our method; though our primary datasets generally perform well across model scales, it is unclear whether enlarging them to encompass more data can provide greater benefits.

6 Conclusion

We introduce COMET, a framework for connecting failure patterns in memory retrieval to retrieval adaptation through diagnosis-guided data construction. Among the five proposed failure patterns, over-fragmentation, memory-query form mismatch and entity mismatch are empirically suggested to be prominent on long-term memory tasks. Adapters for the undiagnosed categories generally do not outperform untargeted adaptation, whereas the three diagnosed categories yield effective, data-efficient adapters across backbones, with gains extending to answering and external benchmarks. The results leave causal pattern-specific attribution and data scaling open, but support a diagnose-then-adapt principle for memory-retrieval adaptation.

7 Reproducibility Statement

There are several portions of the Methods section of the main text which were condensed for brevity, and which are fully detailed in the Appendix section: 1) motivations, formulas and precise constructions for all COMET proxies, and 2) exact training specifications, including hyperparameters and loss formulas, used in the retrieval adaptation section. The Appendix further documents our reproducible procedures for benchmark filtering and preprocessing, all relevant hyperparameters, random seeds, LLM prompts and human audits of LLM outputs. We will release the following as part of a repository accompanying the paper: 1) all code required to reproduce the experimental artifacts; 2) all final result artifacts presented in the paper; and 3) the COMET datasets - in essence, everything needed to reproduce the regression results and the trained adapters. We were unable to include all adapter weights and additional intermediate artifacts due to file size constraints, except for our flagship adapter (OME-750), which is included.

The methodology of the paper relies heavily on LLM detection passes, such as for entities and time expressions present in text. As LLM inference may not be bit-for-bit deterministic, we cannot guarantee exact regeneration of every LLM-derived artifact; however, we will release all code and principal experimental artifacts used in the project in an anonymous repository as supplementary material. The principal tables and statistical results can be reproduced directly from these artifacts. The specific CLI arguments used to run the packaged scripts are listed in the packaged README, including known result fluctuations and deviations.

8 Ethics Statement

This work seeks to advance understanding of the field of agent memory and contribute to the design of more efficient memory systems in the future. Such systems are not ethically neutral: there is an inherent risk in designing systems whose chief aim is to grant LLMs the ability to adopt persistent knowledge of users, and to do so as effectively as possible. In particular, such systems may contribute to the storage of private or otherwise sensitive information of users, and facilitate in users the notion that the memory-augmented LLM literally remembers the user or exhibits an emotional attachment to them. Deployments of persistent memory systems should therefore incorporate informed consent, data minimization, appropriate access controls, and mechanisms through which users can inspect, correct, and delete stored memories. Systems should also communicate clearly that apparent recollection or attachment does not imply human-like memory or emotion.

COMET does not collect any new human-subject data. PersonaMem, LoCoMo, PerLTQA and all other LMEB datasets consist of synthetic or LLM-generated conversational and personal-memory data and do not contain conversational histories collected from real users. REALTALK (Lee et al., 2025) contains authentic human conversational histories; we use only the de-identified version released by its authors under the accompanying data-use conditions and do not attempt to identify or re-identify any participant. The methods introduced in this paper are intended to support the analysis and improvement of retrieval systems, but any deployment based on them remains subject to the privacy and consent obligations associated with the underlying memory data.

9 AI Use Statement

LLMs play an extensive role in this paper's experimental methodology; in particular, they are employed to produce important research artifacts in the following stages - 1) memory and query annotation, including entities, time expressions and relevant spans; and 2) data augmentation for COMET-guided retrieval adaptation. The resulting research artifacts were subject to the validation and human-audit procedures documented in the Appendix. LLMs also assisted in the iterative refinement of some prompts; the authors wrote or substantively revised all final prompts and manually audited samples of the resulting outputs.

Coding agents were used to rapidly scale the implementation and debugging of experimental code. All released code was reviewed and tested by the authors, and the core experimental scripts were reimplemented and validated by the authors against the intended methodology and retained artifacts. AI tools also provided criticism, search assistance, debugging support, and executed routine statistical scripts during the development of the research and manuscript; in the final editing phase, AI tools

helped produce Figure 1 from the authors' design and edit sections of the text for length and clarity. They were not used to draft any substantial manuscript passage in the main text, which remains written by the authors, or to originate the paper's scientific design. The authors have reviewed all AI-generated artifacts and take responsibility for the paper's text, claims, code, and research artifacts.

REFERENCES

P. Chhikara et al. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025. URL https://arxiv.org/abs/2504.19413.

Y. Du et al. PerLTQA: A personal long-term memory dataset for memory classification, retrieval, and synthesis in question answering. *arXiv preprint arXiv:2402.16288*, 2024. URL https://arxiv.org/abs/2402.16288.

D. Firth. Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1):27–38, 1993. doi: 10.1093/biomet/80.1.27. URL https://doi.org/10.1093/biomet/80.1.27.

E. J. Hu et al. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/abs/2106.09685.

X. Hu et al. QueryLink: Leveraging query-memory alignment for long-term reasoning in LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15608–15621, 2026. doi: 10.18653/v1/2026.findings-acl.765. URL https://aclanthology.org/2026.findings-acl.765/.

S. Ji, Y. Li, and B. Hooi. Memory is reconstructed, not retrieved: Graph memory for LLM agents. *arXiv preprint arXiv:2606.06036*, 2026. URL https://arxiv.org/abs/2606.06036.

B. Jiang et al. Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale. *arXiv preprint arXiv:2504.14225*, 2025. URL https://arxiv.org/abs/2504.14225.

C. Latimer et al. Hindsight is 20/20: Building agent memory that retains, recalls, and reflects. *arXiv preprint arXiv:2512.12818*, 2025. URL https://arxiv.org/abs/2512.12818.

D.-H. Lee et al. REALTALK: A 21-day real-world dataset for long-term conversation. *arXiv preprint arXiv:2502.13270*, 2025. URL https://arxiv.org/abs/2502.13270.

K. K. Leung et al. Classifying and addressing the diversity of errors in retrieval-augmented generation systems. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2026. URL https://arxiv.org/abs/2510.13975.

P. Lewis et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2005.11401.

J. Liu et al. SimpleMem: Efficient lifelong memory for LLM agents. *arXiv preprint arXiv:2601.02553*, 2026. URL https://arxiv.org/abs/2601.02553.

A. Maharana et al. 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. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

C. Nie et al. EvoEmbedding: Evolvable representations for long-context retrieval and agentic memory. *arXiv preprint arXiv:2606.21649*, 2026. URL https://arxiv.org/abs/2606.21649.

C. Packer et al. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

J. S. Park et al. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023. URL https://arxiv.org/abs/2304.03442.

P. Rasmussen et al. Zep: A temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025. URL https://arxiv.org/abs/2501.13956.

N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, 2019. URL https://arxiv.org/abs/1908.10084.

M. Tian et al. HiNS: Hierarchical negative sampling for more comprehensive memory retrieval embedding model. *arXiv preprint arXiv:2601.14857*, 2026. URL https://arxiv.org/abs/2601.14857.

D. Wu et al. LongMemEval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024. URL https://arxiv.org/abs/2410.10813.

S. Xiao et al. C-Pack: Packed resources for general Chinese embeddings. *arXiv preprint arXiv:2309.07597*, 2023. URL https://arxiv.org/abs/2309.07597.

W. Xu et al. A-MEM: Agentic memory for LLM agents. *arXiv preprint arXiv:2502.12110*, 2025. URL https://arxiv.org/abs/2502.12110.

B. Yuan et al. Diagnosing retrieval vs. utilization bottlenecks in LLM agent memory. In *MemAgents Workshop at the International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2603.02473.

Y. Zhang et al. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025. URL https://arxiv.org/abs/2506.05176.

X. Zhao et al. LMEB: Long-horizon memory embedding benchmark. *arXiv preprint arXiv:2603.12572*, 2026. URL https://arxiv.org/abs/2603.12572.

APPENDIX A THE COMET FAILURE PATTERNS

Here we give the operational details of COMET's failure representation, including formulas for each proxy and the LLM prompts used (for exact verbatim prompts, see Appendix F). Before calculating the proxies, we employ the data-annotation pipeline for each memory system on each benchmark described below; the main text presents the conceptual overview, while the definitions and system-specific details are collected here.

APPENDIX A.1 FAILURE PATTERNS AS ADAPTATION TARGETS

Each row links a retrieval difficulty to training-query construction over unchanged memories. Patterns can overlap. Measurable proxies support empirical validation through regression models; their formulas are given in the subsection "The COMET proxies" below.

Table A1: COMET failure patterns as adaptation targets.

Pattern	What makes retrieval difficult?	Targeted training examples
Concentration	A useful fact is buried in a long, mixed-topic memory.	Ask about a localized fact within a longer memory.
Over-fragmentation	Identifying a person and finding the requested fact require separate memories.	Link an anchor question to an additional supporting memory.
Memory-query form	A question asks about an approach to learning; the evidence describes concrete behavior.	Ask for an abstract interpretation supported by concrete evidence.
Entity mismatch	A query uses “her mentor” while the memory uses the mentor's name.	Replace a name with a supported referring expression.
Temporal mismatch	A query uses “spring 2022” while the memory uses a specific date.	Change the expression referring to the relevant time.

APPENDIX A.2 PROBLEM FORMULATION

We consider the following **memory retrieval** problem. A conversational history $\mathcal{C} = [S_1, \ldots, S_T]$ consisting of T sessions S_i is mapped to an external memory store $\mathcal{M} = [(k_1, v_1), \ldots, (k_N, v_N)]$ consisting of N key-value pairs (memories), in which k_i represents a **key** used to retrieve the ith memory (potentially a dense embedding, a sparse lexical representation, or multiple embedding vectors) and v_i represents the **value** returned to the answering LLM. Given a natural-language query q, the end-to-end memory system $Mem(\mathcal{C}, q) = a$ generates an answer a through the write-retrieval-utilization pipeline

$$W(\mathcal{C}) = \mathcal{M}; \qquad R(\mathcal{M}, q, k) = \mathcal{V}; \qquad U(L, \mathcal{V}, q) = a,$$

where $W(\mathcal{C})$ represents the **write module** with the conversation history as input and the memory store as output; $R(\mathcal{M}, q, k)$ represents the **retrieval module**, which outputs a ranked collection of k memory values $\mathcal{V} = [v_{i_1}, \ldots, v_{i_k}]$ conditioned on $\mathcal{M}$ and q; and $U(L, \mathcal{V}, q)$ represents the **utilization module**, producing a final answer a via LLM L for query q by utilizing $\mathcal{V}$. (Note that memory values are human-readable texts indexed by memory keys and passed to the answering model after retrieval.)

For a query q with ground-truth answer a^* supported by $\mathcal{C}$, we call a collection of memory values **answer-sufficient** if it contains enough information for an oracle answering model to correctly answer q. An answer-sufficient collection need not be unique or minimal and may include irrelevant memories; we use $\mathcal{V}_q^*$ to denote any such collection.

Based on this, we attribute each incorrect response to the earliest stage of the pipeline at which answer-sufficient information becomes unavailable or fails to be used:

1. **Write failure**: No answer-sufficient collection can be formed from the values in $\mathcal{M}$; necessary information for answering the query has been lost during ingestion.

2. **Retrieval failure**: An answer-sufficient collection $\mathcal{V}_q^*$ exists among the values in $\mathcal{M}$, but the retrieved collection $\mathcal{V}$ is not answer-sufficient.
3. **Utilization failure**: The retrieved collection $\mathcal{V}$ is answer-sufficient, but the LLM L does not utilize it to produce a correct answer.

APPENDIX A.3 BENCHMARK DATA-PROCESSING

In our experiments, we employed two benchmarks specialized for evaluating long-term memory systems: LoCoMo and PersonaMem. Below, we describe what specialized data-processing procedures were used for each benchmark prior to evaluation.

- LoCoMo presents a relatively simple organizational scheme: each query is linked to a specific conversation, and the eligible set of memories for that query are all memories also linked to that conversation. Retrieval is conducted over that eligible set.
- PersonaMem intersperses queries and conversation turns; within a conversation, queries may occur at several timestamps - after all conversation turns, or anywhere in the middle of the conversation. A query's eligible set of memories for retrieval is all memories with a timestamp earlier than or concurrent to the query, within the same conversation, as specified officially by the benchmark.
- **Eligible questions**. On LoCoMo, we exclude category-5 (adversarial) questions, 4 questions without benchmark evidence, and 2 whose evidence IDs do not appear in the conversation, leaving 1,534 of 1,986. On PersonaMem, all 589 questions are eligible. We further exclude write failures from our regression analysis, which are questions deemed to have no eligible ground-truth memory set (as noted below).

APPENDIX A.4 MEMORY-STORE ANNOTATION

- The memory system is used to generate a memory store per the authors' original paper specifications or current GitHub specifications, whichever is newest. For our systems, the specifications are as follows:
  - RAG: No modification to raw benchmark text. Each conversation turn is stored as a distinct memory, in the form "[DATE] SPEAKER_NAME: CONTENT", where DATE is the ISO-8601 absolute date of the turn (year-month-day), SPEAKER_NAME is the name of the speaker labeled by the benchmark, and CONTENT is the raw content of the turn.
  - Mem0: gpt-5-mini at high reasoning effort was used as the memory generation model, as specified by the authors' current GitHub. The system prompt used is the newest version specified by the authors, which involves ADD operations only (no memory merging or deletion). Distilled atomic facts are generated at the session level, and embedded as distinct memories.
  - A-MEM: gpt-5-mini at high reasoning effort was used as the memory generation model. While the authors specified gpt-4o-mini, the model is now outdated and performs significantly worse compared to more modern alternatives; we opt for 5-mini for better comparability with Mem0 and superior performance. For link construction, we use the author-specified all-MiniLM-L6-v2 embedding model (Reimers & Gurevych, 2019); though we later evaluate downstream retrieval with different embedders, such as Qwen3-Embedding-0.6B, altering the embedding model at store construction time would introduce a further methodological confound and introduce unnecessary scope. Instead of creating summarized memories across entire sessions, A-MEM augments raw conversational turns with distilled summaries, concepts, and links to related turns which enable graph-traversal retrieval.
- The memory store is then annotated as follows:
  - Each query q is annotated with a set of **ground-truth memory values** $\mathcal{V}_q$: a set which is **sufficient** (contains enough information to correctly answer the query) and **minimal** (removing any member would render it insufficient). RAG and A-MEM do not require annotation; as their memories are each linked to raw turns, the benchmarks themselves supply the annotation. An LLM is used to annotate such sets for Mem0 (see Prompt A1).

  - For Mem0, we exclude all questions that the LLM judges to be a write failure: no set of memories in the store is sufficient to answer the query. For PersonaMem, which injects user profiles into the system prompt, an additional subset of questions is marked by the LLM to be answerable without memory retrieval. This set (two separate LLM passes annotated 35 questions for Mem0, and 34 for A-MEM) is excluded for Mem0 and A-MEM for comparability across the two specialized memory systems. The no-memory-required questions are excluded from all three systems including RAG during adapter evaluation, with RAG sharing A-MEM's annotations (same turn-level memory), so that all three systems can be evaluated only for questions whose answer requires retrieving a memory.
  - For each query q and memory value v in the entire memory store $\mathcal{M}$, we obtain the following annotations and representations:
    * **Entities** $E(x)$, a list of entities present in a query or memory's raw text x; each entry is represented as a list of co-referring surface forms that represent the same entity.
    * **Time expressions** $T(x)$, a list of temporal expressions (t, t_{start}, t_{end}) with t the raw string, t_{start} the ISO-8601 start time and t_{end} the ISO-8601 end time (if resolvable within given context).
    * **Residual lemmas** $L(x)$, a deterministic set of lemmatized words remaining in text x after removal of entities, time expressions, and stop-words. Document frequencies $\text{df}(l) = \sum_{i=1}^{N} 1\{l \in L(v_i)\}$ are calculated over all memory values $v_1, \ldots, v_N$ for each unique lemma l.
    * **Relevant spans** $S(q, v)$, a list of non-overlapping spans within v which contribute to recognizing v as relevant to q; this is annotated per query-ground-truth-memory pair. We use $\text{len}(S(q, v))$ to denote their total length in characters.
- **Annotation specifications**. The annotations above are performed with GPT-5.6-Sol at high reasoning effort with temperature 1 (the lowest permissible temperature). Entity and time expressions are annotated in the same LLM pass, governed by Prompt A2. Relevant spans are annotated with Prompt A3.

Appendix A.5 The COMET proxies

Following this annotation pipeline, the exact definitions for the COMET proxies are given below.

- **Concentration**, $C(q)$: $C(q) = 1 - \text{avg}_{v \in \mathcal{V}_q} \frac{\text{len}(S(q,v))}{\text{len}(v)}$, one minus the average ratio of relevant span length to total memory value length (in characters). Concentration is higher when the spans establishing the relevance of the ground-truth memories occupy a smaller proportion of their total lengths.
- **Over-fragmentation**, $O(q)$: $O(q) = |\mathcal{V}_q|$. A query has **more over-fragmented memories** if it has a larger set of ground-truth memories. We make two notes regarding this definition:
  - Under uniform random retrieval, a larger set of ground-truth memories leads to a lower expected binary retrieval success, whereas expected Recall@k does not depend on $|\mathcal{V}_q|$ (see the subsection "Over-fragmentation and Recall@k" below). As such, though a significant coefficient for this proxy with binary retrieval failure as the dependent variable is not informative, the same significant coefficient would point to more than a mathematical artifact if reproduced under Recall@k.
  - We note that, though the two are connected, over-fragmentation is not the literal failure mode of "being a multi-hop query"; whether a query requires retrieving multiple memories, i.e. multi-hop, depends on how these memories are stored, and whether disparate pieces of information are kept separate or integrated into a single memory.
- **Memory-query form mismatch**, $M^+(q)$ and $M^-(q)$. Define the following form-similarity function between any memory value-query pair q and v:

$$FS(q, v) = \frac{\sum_{l \in L(q) \cap L(v)} w(l)}{\sum_{l \in L(q)} w(l)}$$

  with $w(l)$ denoting the **smoothed inverse document-frequency weight** $w(l) = \log(\frac{N+1}{\text{df}(l)+1}) + 1$, and N the total number of memories in the query's **candidate universe**

(all stored memories corresponding to the same conversation as the query; for PersonaMem, the query's snapshot containing all preceding memories). This is a crude proxy for whether the query and the memory contain discernibly similar surface forms. As this is measurable over non-ground-truth memories also, we calculate:

- $M^+(q) = 1 - \min_{v \in \mathcal{V}_q} FS(q, v)$, the **attractor memory-query form mismatch** proxy; and
- $M^-(q) = \sum_{v \notin \mathcal{V}_q} FS(q, v)$, the **distractor form pressure** proxy, kept as a non-normalized sum to quantify both the similarity of non-ground-truth memories to the query and the number of such memories.

- **Entity mismatch**, $E^+(q)$ and $E^-(q)$. Define the following entity-similarity function between a query entity e_1 (a list of co-referring surface forms) and a memory value v as

$$ES(e_1, v) = \max_{e \in e_1,\ e' \in \bigcup E(v)} \text{bigram_sim}(e, e')$$

representing the maximum bigram similarity (Jaccard over character bigrams) achievable between a surface form in e_1 and any entity surface form in v. We define $ES(e_1, v) = 0$ when $E(v)$ is empty, and calculate:

- $E^+(q) = 1 - \min_{v \in \mathcal{V}_q} \left(\frac{1}{|E(q)|} \sum_{e \in E(q)} ES(e, v) \right)$, the **attractor entity mismatch** proxy; and
- $E^-(q) = \sum_{v \notin \mathcal{V}_q} \left(\frac{1}{|E(q)|} \sum_{e \in E(q)} ES(e, v) \right)$, the **distractor entity pressure** proxy.

- **Temporal mismatch**, $T^+(q)$ and $T^-(q)$. For a query time expression $\tau = (t, t_{start}, t_{end})$ and memory time expression $\tau' = (t', t'_{start}, t'_{end})$, define

$$TS(\tau, \tau') = \begin{cases} 2, & \text{bigram_sim}(t, t') > 0.5 \text{ and the resolved intervals overlap,} \\ 1, & \text{bigram_sim}(t, t') > 0.5 \text{ otherwise,} \\ 0, & \text{otherwise.} \end{cases}$$

For each $\tau \in T(q)$, let $\tau_v^* \in T(v)$ denote the memory time expression with the greatest bigram similarity to t, and define the query-memory temporal similarity

$$TS(q, v) = \frac{1}{|T(q)|} \sum_{\tau \in T(q)} TS(\tau, \tau_v^*).$$

We define $TS(q, v) = 0$ when $T(v)$ is empty, and calculate:

- $T^+(q) = 1 - \frac{1}{2} \min_{v \in \mathcal{V}_q} TS(q, v)$, the **attractor temporal mismatch** proxy; and
- $T^-(q) = \sum_{v \notin \mathcal{V}_q} TS(q, v)$, the **distractor temporal pressure** proxy.

The entity proxies are defined only when $E(q)$ is non-empty, the temporal proxies only when $T(q)$ is non-empty, and the form proxies only when $L(q)$ is non-empty. These proxies are measured at the benchmark-system level; they are not dependent on the embedder used or the value of k for Recall@k.

Our retrieval metrics (binary and Recall@k) apply only to retrieval-eligible questions, meaning we exclude questions exhibiting **write failures**: queries which the entire candidate universe (memories linked to the same conversation as the query) provides insufficient information for. This is applicable for Mem0 only; we prompt the LLM performing the ground-truth memory annotations described above to return whether a write failure has occurred, and exclude such questions from being eligible for retrieval failures. A retrieval failure is then defined as questions where, under a fixed k, not all members of the ground-truth memory set are retrieved (i.e. Recall@$k < 1$).

APPENDIX A.6 OVER-FRAGMENTATION AND RECALL@k

We give a reference calculation to separate the effect of requiring complete retrieval from the effect of measuring partial retrieval. Let the candidate memory store contain N values, let the ground-truth set contain $g = |\mathcal{V}_q|$ values, and suppose retrieval selects a uniformly random subset of k values without replacement. For each ground-truth value $v \in \mathcal{V}_q$,

$$\Pr(v \text{ is retrieved}) = \frac{k}{N}.$$

By linearity of expectation, the expected Recall@k is therefore

$$\begin{aligned}\mathbb{E}[\text{Recall@}k] &= \frac{1}{g} \sum_{v \in \mathcal{V}_q} \Pr(v \text{ is retrieved}) \\ &= \frac{1}{g}\, g \frac{k}{N} = \frac{k}{N}.\end{aligned}$$

Thus, under uniform random retrieval, expected Recall@k does not depend on the number of ground-truth memories. Complete retrieval has a different dependence. If $k < g$, retrieving every ground-truth value is impossible and the probability is zero. If $k \geq g$,

$$\Pr(\mathcal{V}_q \subseteq R(\mathcal{M}, q, k)) = \frac{\binom{N-g}{k-g}}{\binom{N}{k}} = \prod_{j=0}^{g-1} \frac{k-j}{N-j}.$$

For $k < N$, each additional required ground-truth value multiplies this probability by a factor smaller than one, so complete-retrieval probability decreases as g increases (and is already zero once $g > k$). This is why we use both complete-retrieval success and continuous Recall@k: the former captures the increasing burden of retrieving every fragmented value, while the latter is not mechanically driven by the size of the gold set under this uniform reference model.

Appendix A.7 Implications for memory-system design

The main text adapts the retriever while holding the memory store and the query fixed. The same failure patterns also suggest storage- and query-side modules, some already present in existing systems, that could complement retrieval adaptation. We have not evaluated these modules; they are design directions rather than results, and a system that combines them is a possible direction for future work.

- **Atomic memory units.** Summarizing sessions into atomic facts may reduce concentration pressure, since each stored fact carries little distracting content: concentration is not significantly associated with binary retrieval failure for Mem0 on either benchmark (Tables C9 and C12), whereas it is strongly significant for RAG and A-MEM on PersonaMem (Tables C11 and C13). The trade-off is fragmentation, for which Mem0 shows the strongest results (Section 4.1).
- **Graph-structured memory.** Linking related memories, as A-MEM does, may offset fragmentation by letting retrieval move from a dense-retrieved seed to linked memories. Temporal mismatch is not significant for A-MEM on either benchmark (Tables C10 and C13), which is consistent with linked events aiding temporal retrieval but does not establish it; temporal mismatch is also not significant for RAG on PersonaMem (Table C11).
- **Query decomposition.** For over-fragmentation, a query could be decomposed once into sub-queries that each target a single fact or information need, with the retrieved results fused. Unlike MRAgent (Ji et al., 2026), which iteratively extracts retrieval cues and updates retrieval decisions from accumulated evidence, a one-time decomposition may reduce latency.
- **Implicit-form query expansion.** For memory-query form mismatch, a query could be paraphrased towards the concrete evidence that would answer it, for example rewriting "Is Caroline religious?" as "Has Caroline participated in religious activities?". QueryLink (Hu et al., 2026) generates hypothetical events or intents for a query, but does not require them to avoid restating the answer (e.g. "Caroline is religious"), which may leave the retrieved set unchanged. Decomposition and expansion could also be combined in an agentic loop in which an LLM chooses the rewrite at inference time.
- **Entity and time canonicalization.** Hindsight (Latimer et al., 2025) resolves entity mentions to a global set of canonical entities and normalizes temporal expressions in stored memories,

but does not canonicalize queries. Combining store-side canonicalization with query-side canonicalization, as in QueryLink (Hu et al., 2026), would target entity and temporal mismatch directly.

APPENDIX B DIAGNOSIS-GUIDED ADAPTATION

Here we elaborate on two main aspects of the methodology for COMET-guided retrieval adaptation presented in the main text:

- Construction of COMET-specialist datasets, including summaries of prompts used and construction of contrastive loss triplets;
- Training protocol for all LoRA adapters, including LoRA configuration, optimizer settings and loss used.

Results for these experiments are presented in Appendix D.

APPENDIX B.1 DATASET CONSTRUCTION

As stated in the main text, each COMET-specialist dataset (corresponding to the five COMET categories: concentration, over-fragmentation, memory-query form mismatch, entity mismatch and temporal mismatch) contains 250 examples augmented from the PerLTQA dataset. This is done as follows:

- A coding agent is given the raw PerLTQA dataset file, containing all 5,266 questions in the official train split as well as their designated memories. In this experiment, we used GPT-5.6-Sol at high reasoning effort hosted through the Codex wrapper.
- The agent is prompted to minimally edit existing eligible queries (defined below) to generate examples of each COMET failure mode. Each query must point to unmodified PerLTQA memories. For each query, the agent records its text and the ground-truth memories it points to (alongside other supplementary information, such as entity types, that were used in internal testing only).
- No artificial quotas, such as "query must contain more than three ground-truth memories" or "there must be more than 100 questions exhibiting no entity overlap", were imposed. For a summary of the prompts used for each COMET category, see the table below; for exact prompts, see Appendix F, prompts B1 to B5. All dataset construction was performed in a single coding-agent pass; if fewer than 250 questions were constructed in the first pass, the agent was simply instructed to continue finding eligible examples. The change in each dataset's targeted COMET proxy from the source query to the augmented query is reported in Table D13.
- To avoid confounding the COMET categories, all constructed questions have exactly one ground-truth memory (nearly all original PerLTQA questions are single-memory; 31 of 5,266 have more than one), aside from the over-fragmentation dataset by definition. Furthermore, the agent is explicitly instructed to target only a single COMET failure mode per query.
- All prompts share the following rough scaffold:
  - Select one existing source question and its labelled ground-truth memory ("gold memory"); preserve it as the ground-truth memory of the constructed question
  - As the memory cannot be changed, the model is instructed to select only memories which can support the failure mode without completely reconstructing the query
  - Rewrite the query such that the targeted COMET failure mode is exhibited in a non-artificial or contrived manner, according to detailed guidelines we summarize below
  - The model is instructed to carefully validate that the selected ground-truth memories are the only memories which answer the query, and are sufficient for it; otherwise, it rejects the example
- An additional placebo dataset, using the same construction agent, is constructed at the scale of 1250 queries, from which Placebo-250 (its first 250 queries) and Placebo-750 (a random subset, seed 42) are drawn; it matches the construction agent used and the query augmentation procedure, featuring instructions to perform a generic rewrite of the query that controls the granularity, entities, time expressions and lexical overlap with the memory.

Note that the placebo prompt (B6) has no rejection stage: the placebo set is fixed at the same size as the COMET datasets and every sampled source question is rewritten, whereas the category prompts (B1-B5) reject examples that fail validation. The placebo therefore matches the augmentation procedure but not the filtering rigor of the targeted datasets.

- Finally, as the placebo does not control for data selection, we evaluate a separate selection control for each primary adapter (C-250, O-250, M-250, E-250, T-250 and OME-750). This control uses all source questions from PerLTQA selected by the construction agent, but without augmentation, isolating the effect of whether the selected source question is rewritten; for OME-750, the selection controls for O-250, M-250 and E-250 are concatenated with duplicates kept.

Table B1: Summary of prompts used for each COMET category, and the placebo generic-augmentation dataset, in COMET-guided retrieval adaptation. The COMET category datasets are built with prompts B1 to B5, respectively. The placebo dataset is built with Prompt B6; the answering and judging prompts used for the end-to-end evaluation are Prompts B7 and B8.

COMET category	Eligible memories	Prompt summary
Concentration	Memories which contain several pieces of information that are semantically dissimilar (at least three competing facts in addition to the fact relevant to the query).	Rewrite the original query linked to the chosen memory such that the part of the memory genuinely answering the query must occupy a small portion of its length, e.g. 5%. Do not make the query artificially specific, e.g. naming exact dates, participants or organizations.
Over-fragmentation	Sets of memories (size 2 or more) that contain complementary information for multi-hop reasoning. The size of this set is not limited; the model is not explicitly instructed to use sets of any size.	Choose one existing query as an anchor. Retain its original answer, syntax, vocabulary and structure as far as possible. Replace named persons, events, places, times etc. with a natural description that must be resolved with one additional memory to create a multi-hop question.
Memory-query form mismatch	Any memory-query pair where the query asks about information that is only implicitly expressed in the memory, e.g. "Is Caroline religious?" -> "Caroline painted a church last week."	Rewrite the original query such that the query and the memory represent the same information in different forms: e.g. if the query asks about a personality trait, the memory should express that trait through a concrete action rather than explicitly stating it. The memory must answer the query intelligibly.
Entity mismatch	Any memory which contains entities that can be expressed through different surface forms that can be understood to co-refer to the same entity without outside context.	Rewrite the original query such that all shared entities between the query and the memory are expressed differently in the query and the memory. The differing surface forms must be recognizable as the same entity without any outside context.
Temporal mismatch	Any memory which contains time expressions that can be expressed through different surface forms. Note that in PerLTQA, all memories contain absolute timestamps that are naturally part of the memory.	Rewrite the original query such that the query now has a temporal information need, but expresses the time expression (e.g. absolute timestamp) in the memory through different surface forms: e.g. a coarser form ("Spring 2022" vs. "March 15, 2022"), expressions containing before/during/after, etc.

COMET category	Eligible memories	Prompt summary
Placebo (generic augmentation)	All memories.	Rewrite the original query through altering grammar, phrasing and vocabulary while keeping granularity (level of broadness of the question), temporal expressions, entities, and general lexical overlap to the target memory controlled.

APPENDIX B.2 TRAINING EXAMPLES

Following dataset construction, we follow these procedures to transform the constructed examples into trainable (query, positive example, negative example) triplets for contrastive loss. Note that these procedures apply to training on the original PerLTQA examples themselves as well.

- For each query, use all labeled ground-truth memories as the positive example in training triplets. For over-fragmentation questions where there are multiple such memories, this means that each ground-truth memory can serve as a separate positive example for triplets to be anchored to.
- As is standard in contrastive learning, for each query we mine for negative examples by calculating the cosine similarity between the embedding of the query and all memories in the memory store by the base embedder, then picking the top-50 non-ground-truth (i.e. incorrect) memories; these negatives are shared across all of the query's positive examples, and are the most difficult or easily confused distracting memories as determined by the base embedder. The negatives for OME-750 and COMET-1250 on Qwen3-Embedding-4B and BGE-large-en-v1.5, and for the C-, O-, M- and T-250 specialists and the PerLTQA-250, PerLTQA-1250 and PerLTQA-Full controls on Qwen3-Embedding-4B, were mined once with Qwen3-Embedding-0.6B and reused; all other arms, including every Placebo control, use negatives mined with their own backbone. For OME-750 and COMET-1250, reuse keeps each training file identical across backbones, so that their cross-backbone results evaluate one fixed curriculum. As negatives mined with another model are less matched to the trained backbone, we expect OME-750's retrieval margins over its backbone-mined Placebo-750 and PerLTQA-750 controls, which are positive on both backbones, to be conservative: a seed-42 Qwen3-Embedding-4B retrain with 4B-mined negatives raises both six-cell Recall@10 margins by about 0.3pp (both then significant), although it lowers LMEB nDCG@10 by 0.7pp. We note that 50 is an arbitrary hyperparameter; though we have tested different values, such as 20 or 30 negative examples per query, performance is not especially sensitive to this parameter.
- We treat each query, not each positive example or triplet, as an independent example. For a query q with positive memories P_q and negative memories N_q, the question-normalized pairwise softplus loss is

$$L_q = \frac{1}{|P_q||N_q|} \sum_{p \in P_q} \sum_{n \in N_q} \text{softplus}\left(s\left[\text{cossim}(q,n) - \text{cossim}(q,p)\right]\right).$$

- Here, cossim is cosine similarity between L2-normalized embeddings and the fixed scale is $s = 20$. The overall training loss is the average of L_q across queries, so questions with more gold memories do not receive greater total weight. Negative examples are processed in microbatches with weighting that preserves the full negative-set mean.

APPENDIX B.3 TRAINING CONFIGURATION

- We train five independent specialist adapters, one for each COMET category:
  - entity mismatch;
  - temporal mismatch;
  - memory-query form mismatch;

  - over-fragmentation;
  - concentration.
- Each specialist is trained on its own 250-question constructed curriculum. We further train one OME-750 adapter on the union of the over-fragmentation, memory-query form mismatch and entity mismatch curricula, and one COMET-1250 adapter on the union of all five curricula.
- For each dataset size (250, 750, 1250), we train a matched adapter using the same number of questions from the original PerLTQA dataset, randomly drawn.
- Generic-Hard-250 is a difficulty control: 250 unaugmented PerLTQA training questions sampled uniformly from the hardest Recall@10 quintile under each unadapted backbone (lower Recall@10 first, then worse gold rank), trained like the other 250-question adapters. It is not a matched control and does not enter the bolding rule.
- As described above, the training contract is:
  - 250 questions per specialist;
  - 50 frozen hard negatives per query, shared across its ground-truth memories;
  - All labeled ground-truth memories retained as positive examples;
  - One question-level optimizer update per training question
- The base embedder used is Qwen3-Embedding-0.6B through SentenceTransformers (Reimers & Gurevych, 2019). We replicate these training runs with Qwen3-Embedding-4B and BGE-large-en-v1.5 to test whether the effect is preserved across embedding scales and model families.
- The following LoRA configuration is used:
  - Task type: feature extraction.
  - Rank $r = 8$, $\alpha = 16$.
  - Dropout 0.05.
  - Target "q_proj", "k_proj", "v_proj" and "o_proj" for Qwen3-Embedding; "query", "key", "value" and "attention.output.dense" for BGE-large-en-v1.5.
  - No bias.
  - Training seeds 42, 43, 44.
  - Base model loaded in float16, with maximum sequence length 2048 tokens for Qwen3-Embedding and 512 tokens for BGE-large-en-v1.5; only the LoRA parameters are trained.
  - As Qwen3 is an instruct-aware embedding model, the following generic memory-retrieval instruction is used for the query prefix:

```
Instruct: Given a question, retrieve relevant memory passages that contain evidence needed to answer
the question
Query:
```

- The same instruction prefix is used when training the BGE-large-en-v1.5 adapters, which are evaluated with BGE's standard query prefix ("Represent this sentence for searching relevant passages: "). This does not affect the BGE results substantially; matching the training prefix to the evaluation prefix changes OME-750 by +0.79 points (mean Recall@10 over LoCoMo and PersonaMem) and -0.13 points (LMEB nDCG@10). The reported BGE retrieval results are therefore slightly conservative.
- The following optimizer configuration is used: Adam, with fixed learning rate 2×10^{-5}, no weight decay or learning-rate warmup. Questions are shuffled each epoch.
- Documents receive no instruction prefix. All positives and the 50 frozen negatives contribute to the question-normalized softplus loss defined above. Negatives are processed in memory-safe microbatches, with gradients accumulated before one optimizer update per question; this batching choice does not change the question-level objective or update count.
- The reported specialist and matched PerLTQA-250 checkpoints are fixed at epoch 2, giving 500 question-level updates per adapter. The 750-question adapters (OME-750, PerLTQA-750, Placebo-750) and the 1250-question adapters (COMET-1250, PerLTQA-1250, Placebo-1250) also use epoch 2, giving 1500 and 2500 updates respectively. PerLTQA-Full uses one epoch to expose the model to every eligible source-training question, yielding 5,266 updates.

APPENDIX B.4 DOWNSTREAM EVALUATION

- To assess whether improvements in retrieval metrics yielded by the trained adapters translate to downstream benchmark performance, we evaluate these adapters on our two benchmarks (LoCoMo and PersonaMem) and on all three of our memory systems (RAG, Mem0, A-MEM). As in the main paper, we use recall budget $k = 10$ and base retrieval model Qwen3-Embedding-0.6B.
- We use dense-only retrieval for all three systems to better compare the adapters, as the main difference between them is their dense retrieval capabilities; introducing system-specific retrieval methods like A-MEM graph traversal would confound the comparison.
- For LoCoMo answering, we adopt Mem0's official system prompt for its LoCoMo evaluation (Appendix F, Prompt B7) as A-MEM does not officially release a system prompt. The benchmark data-processing described in Appendix A applies here; eligible memories for a query are all memories from the same conversation, and we grant the answering model access to the $k = 10$ top retrieved memories by cosine similarity.
- For PersonaMem answering, we adopt the official system prompt verbatim (Appendix F, Prompt B7, PersonaMem system prompt) with a slight modification for the model to return only the option label (a, b, c or d, in accordance with PersonaMem being multiple-choice) for easier parsing.
- LoCoMo judging uses the LongMemEval judge prompt (Appendix F, Prompt B8) adapted to output a JSON-formatted object for easier parsing, as the original LoCoMo release did not use LLM judging. We further use F1 to judge LoCoMo answers, as was done in the original paper, and find that the main comparisons (OME-750 vs its controls and the base retriever) replicate under deterministic judging instead of an LLM judge.
- PersonaMem judging uses deterministic parsing of the answer choice outputted.
- The answering model and the judge model used are both gpt-5.6-luna on high reasoning, on temperature 1 (the lowest supported), intended as a balance between cost and capability. A more capable model was not used because this section explicitly intends to measure whether retrieval gains can translate to answering gains with a cost-efficient model that is not necessarily state-of-the-art. We also present results for human audits of LLM judge results in Appendix E to ensure that the assessment is not biased by the judge.

APPENDIX B.5 EXTERNAL COMPARISONS

- We compare against the surrounding literature on memory-retrieval embedder fine-tuning through two main papers: HiNS (Tian et al., 2026) and EvoEmbedding (Nie et al., 2026).
- HiNS fine-tunes bge-small-en-v1.5 (about 33M parameters, roughly 14 times the 2.29M trainable parameters of our 0.6B adapter) with an InfoNCE loss on 201,462 synthetic training samples whose negatives are stratified into hard, medium and easy difficulty; dividing by the 5 to 12 positive messages its generation prompt requests per query suggests roughly 17,000 to 40,000 queries. HiNS evaluates downstream on LoCoMo and PersonaMem with MemoryOS and Mem0, scoring LoCoMo by token F1 and BLEU-1. We do not use it as a quantitative comparison: it does not specify its answering model or retrieval budget, reports single runs without variance, and does not release code, data or model weights.
- EvoEmbedding uses a publicly-released dataset, EvoTrain-180K, which contains 184,137 independent queries. Its embedders are initialized from general language models (Qwen3.5-0.8B for EvoEmbedding-0.8B, Qwen3-4B for EvoEmbedding-4B) rather than from Qwen3-Embedding, and use a memory queue in the encoder architecture itself to continually enrich embeddings with surrounding context.
- EvoEmbedding reports end-to-end answering accuracy on LoCoMo and PersonaMem-32K with a naive RAG pipeline over retrieved segments, a Qwen3-30B-A3B answering model and a GPT-4o-mini judge, at retrieval budgets k of 1, 2, 4, 8 and 16, from single runs. As its answering model, judge and memory units differ from ours, only gains over a shared base embedder are comparable, and only in direction. Their evaluation suite consists of eight benchmarks across the LMEB suite: 5 semantic benchmarks representing general conversational-QA (QASPER, PeerQA, CovidQA, MLDR, ESG-Reports), 3 long-term memory benchmarks (REALTALK, LoCoMo, LongMemEval).
- For comparability to EvoEmbedding's results, we evaluate on the exact same eight benchmarks and calculate overall nDCG@10 using the official LMEB protocol (Zhao et al., 2026).

As shown in Appendix D (Tables D4 and D5), at the smaller scale (0.6B vs. 0.8B), OME-750 slightly exceeds EvoEmbedding on the eight-benchmark average (59.09 vs. 58.97) and on five of eight individual benchmarks, including all three memory benchmarks. At 4B, OME-750 is lower on the eight-benchmark average (60.48 vs. 65.20) but outperforms EvoEmbedding on three of eight benchmarks (LongMemEval, REALTALK and CovidQA), including two of three memory benchmarks.

For supplementary results concerning this section, see Appendix D, where we replicate the primary results presented in the main text across multiple trainer seeds and embedder model scales, including the comparison between COMET adapters and matched PerLTQA adapters.

APPENDIX C SUPPLEMENTARY RESULTS FOR DIAGNOSIS

The main text reports held-out prediction gains (Table 2) and category significance (Figure 2), and summarizes the system-dependent failure profiles. Here we give detailed results for failure diagnosis: 1) a full list of variables used in the regressions and their meanings, along with a pairwise correlation table that demonstrates their relative independence between categories; 2) a replication of the pooled table with Recall@10 as the dependent variable and OLS regression, particularly important for validating over-fragmentation; and 3) summary tables, figures and individual regression tables stratified by benchmark, memory system, choice of embedder and retrieval budget k, including dense-only retrieval for Mem0 and A-MEM. For brevity, we do not reproduce every individual regression table; we present the six tables for our main experimental protocol: LoCoMo and PersonaMem / RAG, Mem0 and A-MEM / Qwen3-Embedding-0.6B / k=10. Each individual regression table features exact coefficients, p-values and goodness-of-fit statistics. We also present Recall@k results for the evaluated memory systems on LoCoMo and PersonaMem for the interested reader, individually across each k and averaged across embedders.

APPENDIX C.1 COMET VARIABLES

Table C1: COMET proxy variables used in the regression models. Each proxy is oriented so that higher values correspond to more adverse retrieval conditions.

Variable	COMET Category	Description
concentration	Concentration	Proxy measuring ratio between relevant and irrelevant information in the ground-truth memory to the query.
fragmentation	Over-fragmentation	Proxy measuring whether information required by the query is over-fragmented between multiple memories.
form_gold_mismatch	Memory-query form	Proxy measuring whether the information need of the query is expressed in similar vocabulary in the ground-truth memories.
entity_gold_mismatch	Entity	Proxy measuring similarity between entities in query and ground-truth memories.
temporal_gold_mismatch	Temporal	Proxy measuring similarity between time expressions in query and ground-truth memories.
form_distractor_pressure	Memory-query form	Proxy measuring whether non-ground-truth memories use similar vocabulary to the query to express distracting information.
entity_distractor_pressure	Entity	Proxy measuring similarity between entities in query and entities in non-ground-truth-memories.
temporal_distractor_pressure	Temporal	Proxy measuring similarity between time expressions in query and time expressions in non-ground-truth memories.

Table C2: Pairwise correlations among the eight COMET proxy variables. Entries are Pearson r values computed on each proxy (normalized to percentiles within the stratum), pooled across the six Qwen3-Embedding-0.6B benchmark/system strata. The largest cross-category absolute correlation is $|r| = 0.258$ (fragmentation with form gold mismatch); the larger within-category correlations are entity gold mismatch with entity distractor pressure ($r = -0.485$) and temporal gold mismatch with temporal distractor pressure ($r = -0.332$). Within-category correlations are expected because the proxies are meant to measure different aspects of the same type of mismatch. The final row gives each proxy's correlation with query-coverage lexical overlap (the fraction of query tokens that also appear in a ground-truth memory, averaged over the query's ground-truth memories, with stop-words kept), computed with the same recipe; its largest absolute value is $|r| = 0.386$ (form gold mismatch).

Proxy	Concentration	Fragmentation	Form gold	Form distractor	Entity gold	Entity distractor	Temporal gold	Temporal distractor
Concentration	1.000	-0.016	0.076	-0.149	-0.115	-0.016	0.005	0.063
Fragmentation	-0.016	1.000	0.258	-0.018	0.128	0.093	0.156	-0.035
Form gold mismatch	0.076	0.258	1.000	-0.259	0.131	-0.129	0.100	-0.034
Form distractor pressure	-0.149	-0.018	-0.259	1.000	-0.003	0.101	-0.061	0.027
Entity gold mismatch	-0.115	0.128	0.131	-0.003	1.000	-0.485	0.095	-0.081
Entity distractor pressure	-0.016	0.093	-0.129	0.101	-0.485	1.000	-0.018	0.103
Temporal gold mismatch	0.005	0.156	0.100	-0.061	0.095	-0.018	1.000	-0.332
Temporal distractor pressure	0.063	-0.035	-0.034	0.027	-0.081	0.103	-0.332	1.000
Query-coverage lexical overlap	0.108	-0.351	-0.386	0.060	-0.295	-0.084	-0.277	0.057

Adding query-coverage lexical overlap as a control to the pooled LoCoMo regressions leaves 12 of the 13 significant category-level associations significant for both binary retrieval failure and Recall@10 (Table C3); only memory-query form mismatch on Mem0 loses significance, consistent with its proxies being themselves built on lexical overlap.

Table C3: Effect of the query-coverage lexical-overlap control on category significance (LoCoMo). For each COMET category, the number of pooled LoCoMo strata (of 3: RAG, Mem0, A-MEM) on which the category's most-significant proxy is significant at $p < .05$, without and with lexical overlap added as a control. Overall, 12 of the 13 category-level associations significant without the control remain significant in both outcomes.

COMET category	Binary failure (no → + overlap)	Recall@10 (no → + overlap)
C (concentration)	$2 \to 2$	$2 \to 2$
O (over-fragmentation)	$3 \to 3$	$3 \to 3$
M (form mismatch)	$3 \to 2$	$3 \to 2$
E (entity mismatch)	$3 \to 3$	$3 \to 3$
T (temporal mismatch)	$2 \to 2$	$2 \to 2$

APPENDIX C.2 RECALL@10 POOLED REGRESSIONS AND OVER-FRAGMENTATION

The table below reports pooled Recall@10 results on Qwen3-Embedding-0.6B across all benchmark/system conditions; the over-fragmentation interpretation and derivation are given in Appendix A.6.

Table C4: Pooled Recall@10 regression results for all benchmarks and systems. Each row is a pooled ordinary-least-squares regression across the three evaluation embedders at $k = 10$, with embedder fixed effects and conversation/persona-cluster bootstrap inference. Controls is the held-out R^2 for the control-only model; Full adds the COMET proxies; OOS gain is the held-out gain in R^2 from adding the proxies beyond controls. RAG is evaluated with dense retrieval; Mem0 and A-MEM are shown using their native retrieval implementations. For dense-only results, see the released paper repository.

Benchmark	System	Mode	n	Clusters	Controls R^2	Full R^2	OOS gain	Over-fragmentation significant?
LoCoMo	RAG	dense	4,602	10	0.122	0.212	+0.073	***
LoCoMo	Mem0	native	4,011	10	0.125	0.248	+0.119	***
LoCoMo	A-MEM	native	4,602	10	0.080	0.168	+0.077	***
PersonaMem	RAG	dense	1,767	20	0.103	0.261	+0.155	ns
PersonaMem	Mem0	native	1,218	20	0.178	0.341	+0.146	*
PersonaMem	A-MEM	native	1,665	20	0.136	0.282	+0.139	ns

APPENDIX C.3 CONTROL VARIABLES

Table C5: Control variables used in the regression models.

Variable	Description
query_words	Number of words in the query.
store_size	Number of memories in the memory store.
gold_words	Average number of words in the ground-truth memories.
query_entity_count	Number of entities in the query.
query_time_count	Number of time expressions in the query.
entity_present	Whether any entities are present in the query.
time_present	Whether any time expressions are present in the query.
form_eligible	Whether the query is eligible for computing the memory-query form proxies.
question_category	Benchmark specific. Categorical variable expressing the category of the query (e.g. temporal, single-hop, multi-hop).
embedder	Embedder fixed effects.

APPENDIX C.4 BENCHMARK RESULTS FOR MEMORY SYSTEMS

As described in the main text, COMET proxy profiles vary by benchmark and memory system; however, Recall@k also differs materially across systems and benchmarks. At $k = 100$, several conditions approach a Recall ceiling and have few remaining binary failures; changes in fit and proxy significance are consequently less reliable. Overall, dedicated memory systems improve over a RAG baseline with date and speaker augmentation (as evaluated in this paper); however, though Mem0 presents the best recall metrics on LoCoMo, it loses a portion of its queries to write failures as annotated by our ground-truth memory annotation pipeline. These queries have had crucial information required for answering them omitted through Mem0's compression of raw sessions into atomic facts. As is consistent with prior literature (Jiang et al., 2025), even dedicated memory systems perform poorly on PersonaMem, with Mem0 even underperforming RAG.

Table C6: Mean Recall@k across embedders by benchmark and system. Each entry is the mean question-level Recall@k, averaged over the three evaluation embedders. Note that A-MEM dense and native results are not directly comparable because A-MEM native retrieval features graph traversal, which inherently requires a greater retrieval budget; this should not be taken as evidence that A-MEM graph traversal provides no benefits. Bolded results are the best-performing system under the benchmark. Mem0 is best on recall metrics but loses a portion of all queries to write failures where query-relevant information is lost from the conversation being summarized into atomic facts, as shown in Table C14.

Benchmark	System	Recall@10	Recall@20	Recall@50	Recall@100
LoCoMo	RAG	0.566	0.650	0.755	0.829
LoCoMo	Mem0 dense	**0.829**	**0.886**	0.948	**0.982**
LoCoMo	Mem0 native	0.819	0.880	**0.952**	0.979
LoCoMo	A-MEM dense	0.636	0.726	0.836	0.905
LoCoMo	A-MEM native	0.573	0.682	0.804	0.888
PersonaMem	RAG	0.589	0.678	0.827	0.958
PersonaMem	Mem0 dense	0.555	0.656	0.807	0.932
PersonaMem	Mem0 native	0.554	0.636	0.797	0.911
PersonaMem	A-MEM dense	**0.604**	**0.695**	**0.847**	0.961
PersonaMem	A-MEM native	0.545	0.661	0.815	**0.964**

APPENDIX C.5 COMET PROFILE VARIATION BETWEEN STRATA

In the main text, we summarized qualitative differences between COMET profiles in different benchmarks, systems and embedders. Here, we provide summary figures representing these differences: overall, between-embedder and between-k differences are fairly minimal, with significant dropoffs at $k = 100$ due to near-complete benchmark saturation for several strata (as shown above). Over-fragmentation, defined as the number of ground-truth memories required to answer the question, remains consistent across k on the rows where it is significant at any k; this suggests that it is not an artifact of a low retrieval budget or, as shown by the Recall@10 results above, a mathematical artifact of more ground-truth memories having lower expected complete recall.

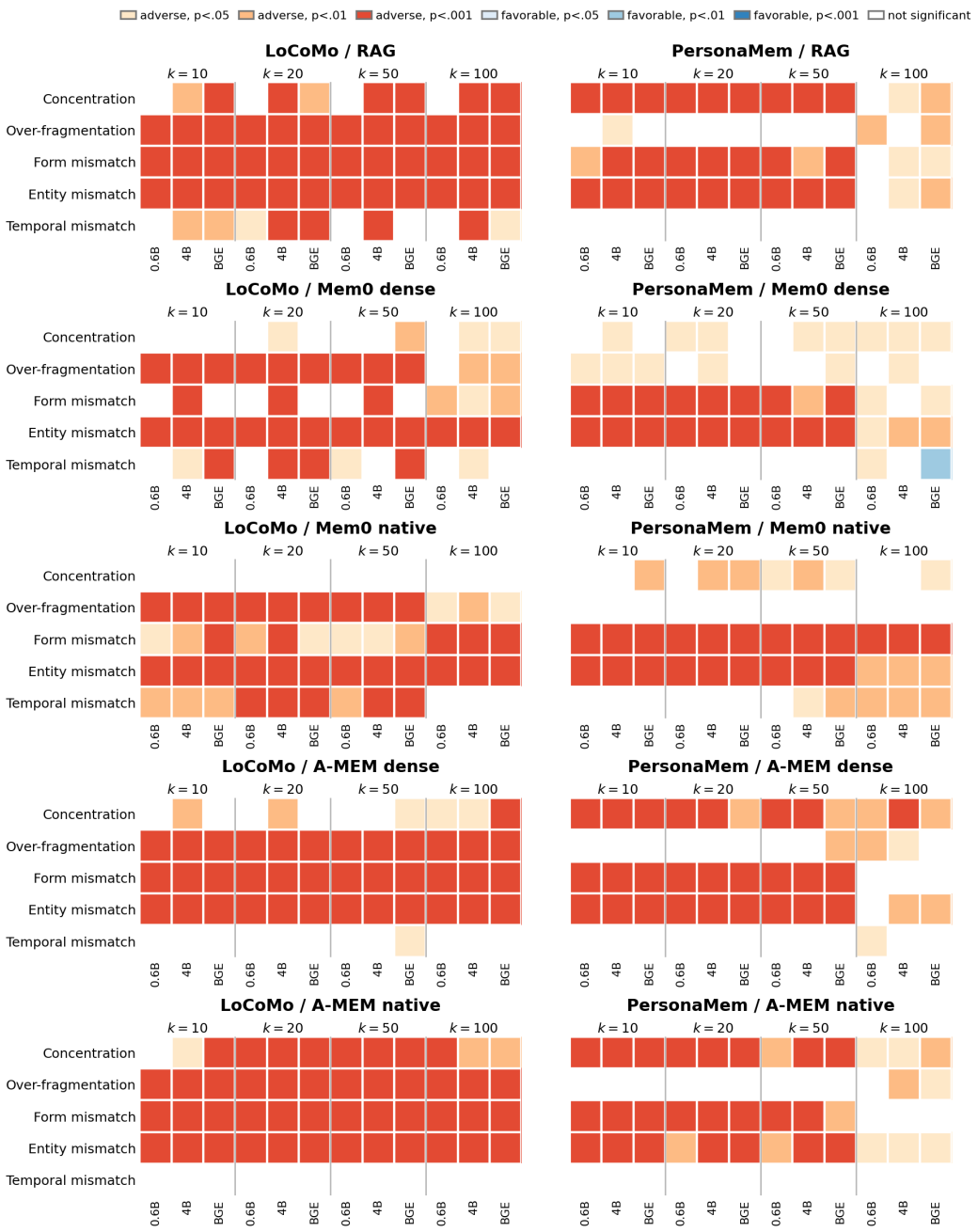

Figure C1: Binary retrieval failure COMET-category significance across strata. Each panel is one benchmark and retrieval configuration (RAG, Mem0 dense, Mem0 native, A-MEM dense, A-MEM native); with the three evaluation embedders, these give the 15 strata per benchmark summarized in Table 2. Columns are grouped by retrieval budget k, with one column for each evaluation embedder within each budget. A square is colored according to the most significant proxy within the category (the lower p-value among its attractor and distractor proxies). Warm colors indicate adverse retrieval associations (positive failure coefficients or negative Recall coefficients), cool colors indicate favorable associations, color intensity indicates $p < .05$, $p < .01$ or $p < .001$, and white indicates $p \geq .05$.

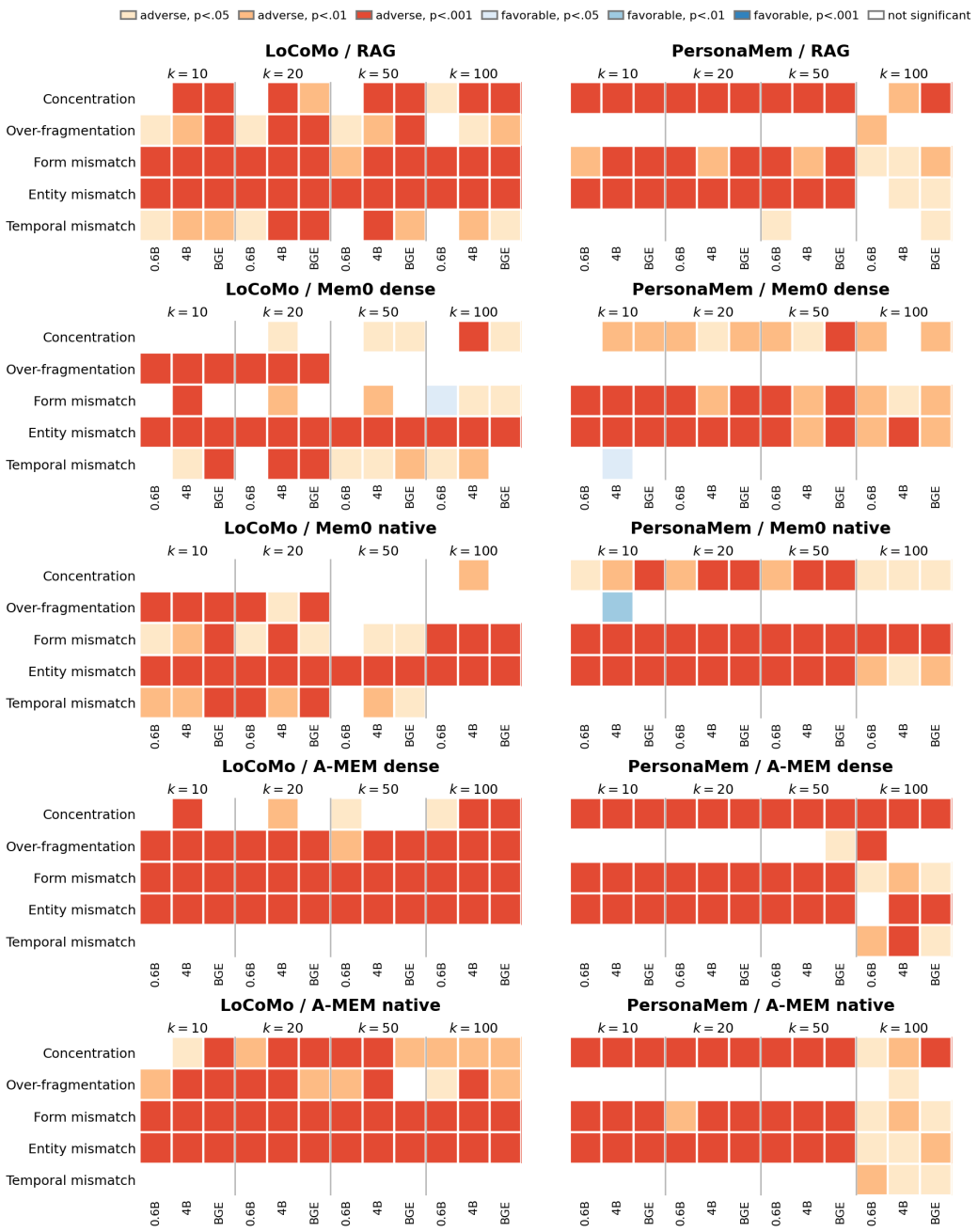

Figure C2: Recall@k COMET-category significance across strata. The grid and category-selection rule follow Figure C1; the coefficients are from the Recall@k OLS models.

Appendix C.6 Sample regression results

Here we report full example regression tables for our primary experimental condition, Qwen3-Embedding-0.6B at k=10 for all memory systems and benchmarks with RAG/dense, Mem0/native and A-MEM/native retrieval.

Table C7: Full $k = 10$, Qwen3-Embedding-0.6B regression results for the primary retrieval conditions. Each condition reports both dependent variables. The coefficient tables show the eight COMET proxies; coefficients, odds ratios where applicable, p-values and held-out goodness-of-fit metrics are shown for RAG/dense, Mem0/native and A-MEM/native. Controls, presence indicators and category effects are released in the repository. Significance markers are * $p < .05$, ** $p < .01$, and *** $p < .001$; coefficients (and odds ratios) are bolded where $p < .05$. The large odds ratio for entity gold mismatch survives Firth correction; the reported coefficients originate from after Firth correction is applied.

Table C8: LoCoMo / RAG / dense regression results.

Outcome	n	Clusters	Controls held-out fit	Full held-out fit	Gain	Bootstrap
Binary retrieval failure	1,534	10	AUC 0.693; pseudo-R^2 0.123	AUC 0.799; pseudo-R^2 0.221	+0.107 AUC; +0.099 pseudo-R^2	2,000
Recall@10	1,534	10	R^2 0.077	R^2 0.159	+0.082 R^2	2,000

Term	Binary coefficient	Binary odds ratio	Binary p	Recall@10 coefficient	Recall@10 p
entity_gold_mismatch	**3.99*****	**53.9*****	<.001	**-0.622*****	<.001
entity_distractor_pressure	**0.00803*****	**1.01*****	<.001	**-0.00126*****	<.001
temporal_gold_mismatch	1.03	2.79	.103	**-0.238***	.040
temporal_distractor_pressure	-0.0314	0.969	.396	0.0042	.529
form_gold_mismatch	**1.12*****	**3.08*****	<.001	**-0.21*****	<.001
form_distractor_pressure	0.00744	1.01	.235	-0.00117	.325
fragmentation	**0.927*****	**2.53*****	<.001	**-0.0223***	.020
concentration	0.306	1.36	.105	-0.0494	.245

Table C9: LoCoMo / Mem0 / native regression results.

Outcome	n	Clusters	Controls held-out fit	Full held-out fit	Gain	Bootstrap
Binary retrieval failure	1,337	10	AUC 0.692; pseudo-R^2 0.110	AUC 0.859; pseudo-R^2 0.316	+0.167 AUC; +0.206 pseudo-R^2	2,000
Recall@10	1,337	10	R^2 0.090	R^2 0.254	+0.163 R^2	2,000

Term	Binary coefficient	Binary odds ratio	Binary p	Recall@10 coefficient	Recall@10 p
entity_gold_mismatch	**4.75*****	**116*****	<.001	**-0.563*****	<.001
entity_distractor_pressure	**0.0275*****	**1.03*****	<.001	**-0.00272*****	<.001
temporal_gold_mismatch	**1.79****	**5.97****	.002	**-0.162****	.002
temporal_distractor_pressure	0.0214	1.02	.184	-0.00213	.088
form_gold_mismatch	**0.982***	**2.67***	.013	**-0.0659***	.048
form_distractor_pressure	0.0406	1.04	.102	-0.00369	.095
fragmentation	**1.11*****	**3.05*****	<.001	**-0.0581*****	<.001
concentration	-0.18	0.835	.580	0.0129	.712

Table C10: LoCoMo / A-MEM / native regression results.

Outcome	n	Clusters	Controls held-out fit	Full held-out fit	Gain	Bootstrap
Binary retrieval failure	1,534	10	AUC 0.689; pseudo-R^2 0.096	AUC 0.781; pseudo-R^2 0.194	+0.092 AUC; +0.098 pseudo-R^2	2,000
Recall@10	1,534	10	R^2 0.064	R^2 0.138	+0.074 R^2	2,000

Term	Binary coefficient	Binary odds ratio	Binary p	Recall@10 coefficient	Recall@10 p
entity_gold_mismatch	**3.18*****	**24*****	<.001	**-0.557*****	<.001
entity_distractor_pressure	**0.00691*****	**1.01*****	<.001	**-0.00128*****	<.001
temporal_gold_mismatch	-0.187	0.83	.687	0.0269	.806
temporal_distractor_pressure	-0.00428	0.996	.436	0.000777	.507
form_gold_mismatch	**1.02*****	**2.77*****	<.001	**-0.196*****	<.001
form_distractor_pressure	**0.00296***	**1***	.011	**-0.000669*****	<.001
fragmentation	**1.04*****	**2.84*****	<.001	**-0.0257****	.003
concentration	0.194	1.21	.052	-0.0129	.504

Table C11: PersonaMem / RAG / dense regression results.

Outcome	n	Clusters	Controls held-out fit	Full held-out fit	Gain	Bootstrap
Binary retrieval failure	589	20	AUC 0.578; pseudo-R^2 0.010	AUC 0.781; pseudo-R^2 0.192	+0.202 AUC; +0.182 pseudo-R^2	2,000
Recall@10	589	20	R^2 -0.007	R^2 0.157	+0.165 R^2	2,000

Term	Binary coefficient	Binary odds ratio	Binary p	Recall@10 coefficient	Recall@10 p
entity_gold_mismatch	**8.25*****	**3.84e+03*****	<.001	**-0.77*****	<.001
entity_distractor_pressure	**0.0675***	**1.07***	.042	-0.00545	.225
temporal_gold_mismatch	-1.08	0.34	.709	-0.0209	.984
temporal_distractor_pressure	0.0314	1.03	.190	-0.00386	.245
form_gold_mismatch	**4.29****	**72.8****	.002	**-0.387****	.002
form_distractor_pressure	**0.0838****	**1.09****	.002	**-0.0116****	.003
fragmentation	0.306	1.36	.451	-0.0232	.836
concentration	**2.76*****	**15.8*****	<.001	**-0.461*****	<.001

Table C12: PersonaMem / Mem0 / native regression results.

Outcome	n	Clusters	Controls held-out fit	Full held-out fit	Gain	Bootstrap
Binary retrieval failure	406	20	AUC 0.674; pseudo-R^2 0.061	AUC 0.834; pseudo-R^2 0.297	+0.161 AUC; +0.236 pseudo-R^2	2,000
Recall@10	406	20	R^2 0.052	R^2 0.221	+0.169 R^2	2,000

Term	Binary coefficient	Binary odds ratio	Binary p	Recall@10 coefficient	Recall@10 p
entity_gold_mismatch	**8.15*****	**3.47e+03*****	<.001	**-0.759*****	<.001
entity_distractor_pressure	**0.0665****	**1.07****	.005	**-0.00974****	.009
temporal_gold_mismatch	9.88	1.96e+04	.453	-0.25	.123
temporal_distractor_pressure	-0.00217	0.998	.977	0.0101	.379
form_gold_mismatch	**14.1*****	**1.35e+06*****	<.001	**-0.756*****	<.001
form_distractor_pressure	**0.297*****	**1.35*****	<.001	**-0.0425*****	<.001
fragmentation	0.248	1.28	.410	0.0512	.134
concentration	1.13	3.09	.060	**-0.219***	.015

Table C13: PersonaMem / A-MEM / native regression results.

Outcome	n	Clusters	Controls held-out fit	Full held-out fit	Gain	Bootstrap
Binary retrieval failure	555	20	AUC 0.622; pseudo-R^2 0.046	AUC 0.762; pseudo-R^2 0.155	+0.140 AUC; +0.109 pseudo-R^2	2,000
Recall@10	555	20	R^2 0.049	R^2 0.156	+0.107 R^2	2,000

Term	Binary coefficient	Binary odds ratio	Binary p	Recall@10 coefficient	Recall@10 p
entity_gold_mismatch	**4.76*****	**117*****	<.001	**-0.646*****	<.001
entity_distractor_pressure	0.0554	1.06	.057	-0.00776	.184
temporal_gold_mismatch	-1.53	0.217	.375	0.193	.478
temporal_distractor_pressure	0.00883	1.01	.655	-0.00241	.500
form_gold_mismatch	**3.51*****	**33.5*****	<.001	**-0.39*****	<.001
form_distractor_pressure	**0.0533*****	**1.05*****	<.001	**-0.00957****	.002
fragmentation	0.3	1.35	.348	-0.0186	.806
concentration	**2.08*****	**7.97*****	<.001	**-0.412*****	<.001

Full regression results are released in the paper repository as JSON files.

Table C14: Original Mem0 write-failure annotations and native Recall@10. Write-failure counts use the complete original GPT-5.6-Sol ground-truth selection annotations, not the 50-question audit sample. An insufficient label means that the complete Mem0 candidate-memory universe was judged insufficient to answer the query. The rate is computed over all annotated benchmark questions. Other excluded records are unanswerable for LoCoMo and no_memory_required for PersonaMem. Native Mem0 Recall@10 is the full-evaluation mean across the three evaluation embedders.

Benchmark	Annotated questions	Insufficient	Write-failure rate	Other excluded	Native Mem0 Recall@10
LoCoMo	1,540	201	13.05%	2	0.819
PersonaMem	589	148	25.13%	35	0.554

APPENDIX D SUPPLEMENTARY RESULTS - COMET-GUIDED RETRIEVAL ADAPTATION

This appendix reports the complete adaptation results behind main-text Tables 3-5: per-benchmark retrieval for each backbone, per-dataset LMEB results, the paired cluster bootstrap tests behind the bolding in the main text, the selection controls, end-to-end answering, the proxy shift of the constructed datasets, case studies, and gains by proxy stratum. Unless stated otherwise, cells are means over trainer seeds 42-44, and the macro column additionally lists the per-seed values (seed 42 / 43 / 44); the base embedder is evaluated once. Per-cell, per-seed and per-contrast results, including bootstrap confidence intervals, are released in the repository.

APPENDIX D.1 RETRIEVAL RESULTS BY BACKBONE

Table D1: Retrieval results, Qwen3-Embedding-0.6B. Mean question Recall@10 (%) under dense retrieval at $k = 10$ over the frozen stores, averaged over the three memory systems for each benchmark; the macro is the unweighted mean over the six benchmark-system cells. COMET adapters and their macros are bolded where the adapter is significantly above at least one matched control (PerLTQA and Placebo of the same size) in at least two of three seeds under the paired cluster bootstrap of Table D7, the rule used for main-text Table 3. Generic-Hard-250 trains on 250 unaugmented PerLTQA questions sampled from the hardest Recall@10 quintile under each unadapted backbone; it is a difficulty control and does not enter the bolding rule. Values are computed from unrounded cells, so differences from Table 3 can appear in the last digit. The Mem0/PersonaMem cell is scored on the 401 questions evaluated for every run, here and in all retrieval tables and bootstrap tests.

Training data	LoCoMo	PersonaMem	Macro (seed 42 / 43 / 44)
Base	66.13	59.98	63.06
PerLTQA-250	78.03	58.91	68.47 (68.49 / 68.34 / 68.58)
Placebo-250	78.51	57.87	68.19 (68.14 / 68.03 / 68.39)
Generic-Hard-250	78.28	56.05	67.17 (67.16 / 67.39 / 66.96)
C-250	76.34	59.58	67.96 (68.01 / 67.99 / 67.89)
O-250	78.62	59.70	**69.16 (69.44 / 69.00 / 69.04)**
M-250	78.64	61.86	**70.25 (70.30 / 70.13 / 70.32)**
E-250	78.55	60.32	**69.43 (69.23 / 69.50 / 69.57)**
T-250	76.83	58.78	67.80 (67.78 / 67.57 / 68.07)
PerLTQA-750	76.49	58.86	67.68 (67.78 / 67.61 / 67.64)
Placebo-750	77.46	60.55	69.01 (68.93 / 68.85 / 69.24)
OME-750	78.73	62.78	**70.76 (70.28 / 70.77 / 71.22)**
PerLTQA-1250	77.54	57.28	67.41 (66.95 / 67.24 / 68.03)
Placebo-1250	76.74	59.99	68.37 (67.99 / 68.83 / 68.27)
COMET-1250	77.25	62.43	**69.84 (70.28 / 69.58 / 69.66)**
PerLTQA-Full	74.06	57.24	65.65 (65.51 / 65.88 / 65.55)

Table D2: Retrieval results, Qwen3-Embedding-4B. As Table D1.

Training data	LoCoMo	PersonaMem	Macro (seed 42 / 43 / 44)
Base	61.15	56.38	58.77
PerLTQA-250	79.25	57.03	68.14 (68.38 / 67.99 / 68.07)
Placebo-250	80.18	57.99	69.08 (68.67 / 68.99 / 69.59)
Generic-Hard-250	77.85	53.75	65.80 (65.88 / 65.08 / 66.44)
C-250	71.88	59.47	65.67 (65.32 / 65.93 / 65.77)
O-250	77.43	58.20	67.82 (67.53 / 68.17 / 67.75)
M-250	78.66	59.72	**69.19 (69.29 / 69.10 / 69.18)**
E-250	78.71	60.65	**69.68 (69.90 / 69.58 / 69.54)**
T-250	77.41	58.66	68.04 (68.01 / 67.66 / 68.44)
PerLTQA-750	81.21	59.51	70.36 (70.96 / 69.99 / 70.14)
Placebo-750	80.79	60.05	70.42 (70.20 / 70.43 / 70.63)
OME-750	80.48	63.40	**71.94 (71.79 / 72.38 / 71.65)**
PerLTQA-1250	82.93	59.44	71.18 (70.60 / 71.83 / 71.12)
Placebo-1250	81.69	59.57	70.63 (70.28 / 71.03 / 70.58)
COMET-1250	81.16	62.91	**72.04 (71.98 / 72.13 / 72.01)**
PerLTQA-Full	79.93	57.21	68.57 (69.23 / 67.16 / 69.33)

Table D3: Retrieval results, BGE-large-en-v1.5. As Table D1. Generic-Hard-250 at trainer seed 42 is markedly lower than at seeds 43 and 44; repeated audits of this run found no fault, and it is reported unchanged.

Training data	LoCoMo	PersonaMem	Macro (seed 42 / 43 / 44)
Base	74.72	58.12	66.42
PerLTQA-250	76.93	58.17	67.55 (67.61 / 67.56 / 67.49)
Placebo-250	77.77	58.21	67.99 (68.13 / 67.99 / 67.84)
Generic-Hard-250	76.53	53.14	64.84 (61.90 / 66.18 / 66.43)
C-250	76.48	60.71	**68.60 (68.56 / 68.76 / 68.47)**
O-250	77.81	61.25	**69.53 (69.60 / 69.55 / 69.44)**
M-250	76.80	62.35	**69.57 (69.53 / 69.68 / 69.51)**
E-250	78.01	60.47	**69.24 (69.37 / 69.21 / 69.15)**
T-250	76.36	59.69	68.02 (67.94 / 68.07 / 68.06)
PerLTQA-750	78.69	59.04	68.87 (68.72 / 68.87 / 69.00)
Placebo-750	78.23	59.06	68.64 (68.75 / 68.70 / 68.48)
OME-750	79.23	62.29	**70.76 (70.81 / 70.59 / 70.87)**
PerLTQA-1250	79.42	59.09	69.26 (69.32 / 69.37 / 69.08)
Placebo-1250	78.21	61.75	69.98 (69.91 / 70.30 / 69.73)
COMET-1250	78.71	62.04	**70.37 (70.60 / 70.34 / 70.18)**
PerLTQA-Full	78.39	61.74	70.06 (70.02 / 70.05 / 70.11)

APPENDIX D.2 LMEB RESULTS BY BACKBONE

Table D4: LMEB nDCG@10, Qwen3-Embedding-0.6B. nDCG@10 (%) on the eight LMEB datasets and their equal-weight macro, following the official protocol. COMET adapters and their macros are bolded where the adapter is significantly above at least one matched control in at least two of three seeds under the LMEB bootstrap of Table D8, as in Table D1. EvoEmbedding is as reported by Nie et al. (2026), who report a single model per scale; its 184,137 training queries compare with our 250-5,266. As a comparability check, our bases match the published ones closely (Qwen3-Embedding-0.6B 50.87 against 50.9; Qwen3-Embedding-4B 48.39 against 48.0).

Training data	ESG-Reports	LoCoMo	LongMemEval	REALTALK	QASPER	PeerQA	CovidQA	MLDR	Macro (seed 42 / 43 / 44)
Base	37.46	34.11	73.09	37.34	46.91	27.06	74.34	76.61	50.87
PerLTQA-250	43.88	45.79	77.66	46.43	50.34	29.16	77.13	77.44	55.98 (55.89 / 56.22 / 55.84)
Placebo-250	43.29	45.99	76.68	46.77	51.00	29.04	78.27	78.00	56.13 (56.05 / 55.87 / 56.47)
Generic-Hard-250	44.92	47.02	78.68	46.80	50.20	28.13	78.21	79.14	56.64 (56.57 / 56.58 / 56.76)
C-250	41.74	43.93	75.88	45.24	50.42	28.13	77.25	77.16	54.97 (54.63 / 55.25 / 55.04)
O-250	42.32	48.10	80.05	46.61	50.97	29.02	79.47	78.91	**56.93 (57.25 / 56.60 / 56.94)**
M-250	51.41	48.49	76.60	46.65	50.87	29.79	76.43	79.65	**57.49 (57.65 / 57.55 / 57.26)**
E-250	43.90	49.00	79.48	46.79	50.77	28.95	77.97	79.51	**57.05 (56.74 / 57.24 / 57.17)**
T-250	37.96	42.38	79.69	44.74	50.98	28.99	77.94	77.66	55.04 (54.84 / 55.06 / 55.24)
PerLTQA-750	44.91	45.93	81.74	47.40	51.55	30.02	78.36	79.99	57.49 (58.09 / 56.95 / 57.42)
Placebo-750	45.01	45.62	80.71	47.21	51.17	29.63	78.56	79.28	57.15 (56.88 / 57.42 / 57.15)
OME-750	47.29	53.31	82.35	47.89	51.75	29.86	79.46	80.83	**59.09 (58.96 / 59.05 / 59.27)**
PerLTQA-1250	42.96	45.16	82.05	47.09	52.13	29.95	79.01	79.29	57.20 (56.45 / 57.09 / 58.08)
Placebo-1250	44.57	44.57	80.69	46.86	52.14	30.58	78.57	79.84	57.23 (57.15 / 57.32 / 57.21)
COMET-1250	44.73	49.98	82.05	46.54	51.43	28.96	79.51	78.19	57.67 (57.83 / 57.48 / 57.70)
PerLTQA-Full	44.23	44.64	81.79	46.18	48.88	29.18	76.15	76.99	56.01 (56.17 / 55.78 / 56.07)
EvoEmbedding-0.8B	57.9	48.8	66.7	45.3	61.7	38.0	75.6	78.1	58.97

Table D5: LMEB nDCG@10, Qwen3-Embedding-4B. As Table D4. EvoEmbedding-4B is as reported by Nie et al. (2026).

Training data	ESG-Reports	LoCoMo	LongMemEval	REALTALK	QASPER	PeerQA	CovidQA	MLDR	Macro (seed 42 / 43 / 44)
Base	41.59	30.08	50.04	34.68	49.59	27.35	76.42	77.36	48.39
PerLTQA-250	49.94	46.37	74.53	45.86	52.48	29.29	79.10	80.54	57.26 (57.45 / 57.35 / 56.98)
Placebo-250	51.37	46.51	74.54	46.35	53.66	30.17	79.75	81.61	58.00 (57.70 / 57.98 / 58.31)
Generic-Hard-250	52.70	39.08	73.75	45.28	52.49	29.94	79.93	83.75	57.12 (57.06 / 56.71 / 57.58)
C-250	45.61	39.23	67.83	41.32	51.57	27.83	78.68	79.67	53.97 (53.52 / 54.50 / 53.88)
O-250	47.43	46.04	77.85	46.58	51.94	28.75	80.03	80.26	57.36 (56.94 / 57.51 / 57.63)
M-250	55.79	45.92	75.79	47.42	51.78	29.25	77.41	81.53	58.11 (58.16 / 57.92 / 58.25)
E-250	51.74	50.49	82.27	46.73	51.99	28.27	79.53	81.69	**59.09 (59.57 / 58.93 / 58.77)**
T-250	44.57	42.65	78.61	44.04	52.31	28.88	80.38	79.71	56.39 (56.29 / 56.15 / 56.75)
PerLTQA-750	52.21	50.19	84.72	48.83	52.49	30.61	80.38	83.18	60.33 (60.97 / 60.08 / 59.93)
Placebo-750	52.88	49.86	83.41	49.12	51.67	29.79	79.92	82.16	59.85 (59.87 / 59.99 / 59.69)
OME-750	57.07	53.92	85.38	50.05	51.80	28.33	77.62	79.68	60.48 (60.74 / 60.40 / 60.30)
PerLTQA-1250	50.91	52.52	85.76	48.87	54.16	31.71	81.99	81.75	60.96 (59.94 / 61.54 / 61.39)
Placebo-1250	53.03	52.27	85.89	49.73	52.96	31.00	81.55	82.73	61.15 (60.40 / 61.45 / 61.58)
COMET-1250	50.44	53.28	86.34	49.65	51.21	29.21	78.60	78.36	59.63 (59.74 / 59.50 / 59.67)
PerLTQA-Full	52.94	51.52	85.31	50.38	51.89	30.47	82.03	82.59	60.89 (60.70 / 60.92 / 61.05)
EvoEmbedding-4B	66.0	61.7	78.8	49.2	66.9	41.1	77.6	80.6	65.20

Table D6: LMEB nDCG@10, BGE-large-en-v1.5. As Table D4. No EvoEmbedding model exists at this scale.

Training data	ESG-Reports	LoCoMo	LongMemEval	REALTALK	QASPER	PeerQA	CovidQA	MLDR	Macro (seed 42 / 43 / 44)
Base	40.48	42.52	75.20	45.74	47.76	29.22	72.94	79.56	54.18
PerLTQA-250	41.13	44.96	76.81	48.41	48.80	29.23	73.62	82.23	55.65 (55.69 / 55.70 / 55.55)
Placebo-250	42.65	46.78	76.67	49.45	49.13	29.67	74.05	82.95	56.42 (56.35 / 56.38 / 56.53)
Generic-Hard-250	46.80	46.39	75.37	48.57	46.91	28.15	71.84	81.66	55.71 (55.83 / 55.74 / 55.56)
C-250	42.94	46.20	75.31	46.51	48.41	30.99	74.84	80.88	55.76 (55.76 / 55.88 / 55.64)
O-250	42.86	47.10	76.55	48.38	49.44	30.87	74.59	82.50	**56.54 (56.68 / 56.42 / 56.51)**
M-250	45.30	44.67	75.50	47.54	45.95	29.25	73.19	81.23	55.33 (55.21 / 55.40 / 55.37)
E-250	46.05	49.63	77.60	49.59	48.40	30.14	75.05	82.99	**57.43 (57.52 / 57.44 / 57.34)**
T-250	39.59	45.31	77.36	47.84	49.74	30.55	74.94	80.53	55.73 (55.60 / 55.69 / 55.91)
PerLTQA-750	45.71	50.19	77.59	50.19	50.10	29.87	75.05	81.26	57.50 (57.56 / 57.32 / 57.61)
Placebo-750	44.93	49.05	77.95	49.07	49.10	29.84	74.68	80.77	56.92 (57.07 / 57.09 / 56.61)
OME-750	47.89	52.27	77.97	50.13	48.41	28.34	74.31	82.97	**57.79 (57.94 / 57.74 / 57.68)**
PerLTQA-1250	44.16	48.98	77.39	50.45	50.69	30.52	75.34	81.78	57.41 (57.32 / 57.73 / 57.19)
Placebo-1250	45.65	47.88	77.16	49.26	48.70	30.13	73.82	82.48	56.89 (56.99 / 56.82 / 56.84)
COMET-1250	45.76	51.47	78.12	50.08	47.86	28.31	74.53	82.70	57.35 (57.61 / 57.38 / 57.07)
PerLTQA-Full	44.12	47.65	77.34	49.14	49.46	29.54	74.07	81.32	56.58 (56.37 / 56.92 / 56.46)

APPENDIX D.3 PAIRED CLUSTER BOOTSTRAP TESTS

All tests are paired cluster bootstraps with 2,000 replicates, run separately for each trainer seed. For retrieval, questions are paired within each benchmark-system cell, clusters (LoCoMo conversations, PersonaMem personas) are resampled within each cell, and the six cell differences are averaged into the macro in every replicate. For LMEB, queries are paired within each task, clusters are the source conversation or document, and tasks are averaged within dataset and datasets into the eight-dataset macro. Each cell reports the mean over seeds of the macro difference (treated minus control, percentage points) and, in parentheses, the number of seeds in which the difference is significant at $p < .05$ (two-sided) with the same sign as the mean. Main-text bolding (Tables 3 and 5) and Tables D1-D6 treat an adapter as supported when it is significantly above a matched control in at least two of three seeds.

Table D7: Retrieval bootstrap contrasts. Differences in six-cell macro Recall@10. Cells: mean difference over seeds (number of seeds significant with that sign).

Adapter	Control	Qwen3-Embedding-0.6B	Qwen3-Embedding-4B	BGE-large-en-v1.5
C-250	PerLTQA-250	−0.51 (0/3)	−2.47 (3/3)	**+1.04 (3/3)**
C-250	Placebo-250	−0.23 (0/3)	−3.41 (3/3)	+0.61 (1/3)
O-250	PerLTQA-250	+0.69 (1/3)	−0.33 (1/3)	**+1.98 (3/3)**
O-250	Placebo-250	**+0.97 (2/3)**	−1.26 (2/3)	**+1.54 (3/3)**
M-250	PerLTQA-250	**+1.78 (3/3)**	**+1.05 (3/3)**	**+2.02 (3/3)**
M-250	Placebo-250	**+2.06 (3/3)**	+0.11 (0/3)	**+1.59 (3/3)**
E-250	PerLTQA-250	**+0.96 (2/3)**	**+1.53 (3/3)**	**+1.69 (3/3)**
E-250	Placebo-250	**+1.24 (3/3)**	+0.59 (1/3)	**+1.26 (3/3)**
T-250	PerLTQA-250	−0.66 (1/3)	−0.11 (0/3)	+0.47 (0/3)
T-250	Placebo-250	−0.38 (0/3)	−1.04 (2/3)	+0.04 (0/3)
OME-750	PerLTQA-750	**+3.08 (3/3)**	**+1.58 (2/3)**	**+1.89 (3/3)**
OME-750	Placebo-750	**+1.75 (3/3)**	**+1.52 (3/3)**	**+2.11 (3/3)**
OME-750	COMET-1250	**+0.92 (2/3)**	−0.10 (0/3)	+0.38 (1/3)
COMET-1250	PerLTQA-1250	**+2.43 (3/3)**	+0.85 (1/3)	**+1.12 (3/3)**
COMET-1250	Placebo-1250	**+1.47 (2/3)**	**+1.41 (3/3)**	+0.39 (0/3)
C-250	Generic-Hard-250	+0.79 (1/3)	−0.12 (0/3)	**+3.76 (3/3)**
O-250	Generic-Hard-250	**+1.99 (3/3)**	**+2.02 (3/3)**	**+4.69 (3/3)**
M-250	Generic-Hard-250	**+3.08 (3/3)**	**+3.39 (3/3)**	**+4.74 (3/3)**
E-250	Generic-Hard-250	**+2.26 (3/3)**	**+3.88 (3/3)**	**+4.41 (3/3)**
T-250	Generic-Hard-250	+0.64 (1/3)	**+2.24 (3/3)**	**+3.19 (3/3)**

Table D8: LMEB bootstrap contrasts. Differences in eight-dataset macro nDCG@10. Cells: mean difference over seeds (number of seeds significant with that sign).

Adapter	Control	Qwen3-Embedding-0.6B	Qwen3-Embedding-4B	BGE-large-en-v1.5
C-250	PerLTQA-250	−1.01 (3/3)	−3.30 (3/3)	+0.11 (0/3)
C-250	Placebo-250	−1.16 (3/3)	−4.03 (3/3)	−0.66 (1/3)
O-250	PerLTQA-250	**+0.95 (2/3)**	+0.10 (0/3)	**+0.89 (3/3)**
O-250	Placebo-250	**+0.80 (2/3)**	−0.64 (1/3)	+0.12 (0/3)
M-250	PerLTQA-250	**+1.50 (3/3)**	+0.85 (1/3)	−0.32 (0/3)
M-250	Placebo-250	**+1.36 (2/3)**	+0.12 (0/3)	−1.09 (3/3)
E-250	PerLTQA-250	**+1.07 (3/3)**	**+1.83 (3/3)**	**+1.78 (3/3)**
E-250	Placebo-250	**+0.92 (3/3)**	**+1.09 (2/3)**	**+1.01 (3/3)**
T-250	PerLTQA-250	−0.94 (3/3)	−0.87 (2/3)	+0.08 (0/3)
T-250	Placebo-250	−1.09 (3/3)	−1.60 (3/3)	−0.69 (3/3)
OME-750	PerLTQA-750	**+1.60 (2/3)**	+0.16 (0/3)	+0.29 (0/3)
OME-750	Placebo-750	**+1.94 (3/3)**	+0.63 (0/3)	**+0.86 (2/3)**
OME-750	COMET-1250	**+1.42 (3/3)**	**+0.85 (2/3)**	+0.43 (1/3)
COMET-1250	PerLTQA-1250	+0.47 (1/3)	−1.32 (2/3)	−0.06 (0/3)
COMET-1250	Placebo-1250	+0.44 (0/3)	−1.51 (2/3)	+0.47 (0/3)
C-250	Generic-Hard-250	−1.67 (3/3)	−3.15 (3/3)	+0.05 (0/3)

Adapter	Control	Qwen3-Embedding-0.6B	Qwen3-Embedding-4B	BGE-large-en-v1.5
O-250	Generic-Hard-250	+0.29 (1/3)	+0.24 (0/3)	**+0.83 (2/3)**
M-250	Generic-Hard-250	+0.85 (1/3)	**+1.00 (2/3)**	−0.38 (0/3)
E-250	Generic-Hard-250	+0.41 (1/3)	**+1.97 (3/3)**	**+1.72 (3/3)**
T-250	Generic-Hard-250	−1.60 (3/3)	−0.72 (1/3)	+0.02 (0/3)

APPENDIX D.4 SELECTION CONTROLS

Each selection control trains on the source PerLTQA questions that the construction agent selected for a COMET category, without augmentation (Appendix B), so that the difference from the augmented adapter isolates the rewrite. Selection controls are trained for Qwen3-Embedding-0.6B on trainer seeds 42-44.

Table D9: Augmented adapters against their selection controls, Qwen3-Embedding-0.6B. Retrieval is six-cell macro Recall@10 and LMEB is eight-dataset macro nDCG@10 (%), both three-seed means; the difference columns give the mean and per-seed differences (seed 42 / 43 / 44). Differences are bolded where the augmented adapter is significantly above its selection control in at least two of three seeds (paired cluster bootstrap, as in Table D7).

Adapter	Retrieval: augmented	Retrieval: selection control	Retrieval difference	LMEB: augmented	LMEB: selection control	LMEB difference
C-250	67.96	67.99	−0.03 (+0.00 / −0.15 / +0.07)	54.97	55.36	−0.39 (−0.64 / −0.34 / −0.19)
O-250	69.16	66.41	**+2.75 (+3.14 / +2.77 / +2.32)**	56.93	52.97	**+3.97 (+4.28 / +3.67 / +3.95)**
M-250	70.25	67.69	**+2.56 (+2.77 / +2.50 / +2.40)**	57.49	54.71	**+2.78 (+2.94 / +2.75 / +2.65)**
E-250	69.43	67.11	**+2.32 (+2.11 / +2.31 / +2.56)**	57.05	54.37	**+2.68 (+2.56 / +2.95 / +2.53)**
T-250	67.80	67.70	+0.10 (+0.04 / +0.02 / +0.26)	55.04	54.62	+0.42 (+0.35 / +0.52 / +0.40)
OME-750	70.76	67.53	**+3.23 (+2.65 / +3.27 / +3.75)**	59.09	55.56	**+3.53 (+2.91 / +3.66 / +4.02)**

APPENDIX D.5 END-TO-END ANSWERING

End-to-end answering uses Qwen3-Embedding-0.6B adapters with dense retrieval at $k = 10$ and gpt-5.6-luna as answering model (Appendix B). Accuracy is the share of all questions in a cell answered correctly, with unanswered questions counted as incorrect; LoCoMo and PersonaMem are each averaged over the three memory systems, and the macro is the unweighted mean of the six benchmark-system cells.

Table D10: End-to-end answering accuracy, LLM judge (%). LoCoMo answers are judged by the LLM judge (Prompt B8); PersonaMem answers are scored by exact match of the selected option.

Retriever	LoCoMo	PersonaMem	Macro (seed 42 / 43 / 44)
Base	55.48	65.93	60.70
PerLTQA-750	62.77	66.55	64.66 (64.87 / 64.74 / 64.38)
Placebo-750	63.54	66.31	64.93 (64.84 / 65.32 / 64.62)
OME-750	64.05	66.29	65.17 (65.33 / 65.32 / 64.86)
COMET-1250	63.10	66.55	64.82 (65.55 / 64.25 / 64.68)

Table D11: End-to-end answering accuracy, deterministic scoring (%). LoCoMo answers are scored by token-level F1 against the benchmark answer; PersonaMem answers are scored by exact match of the selected option.

Retriever	LoCoMo	PersonaMem	Macro (seed 42 / 43 / 44)
Base	46.59	65.93	56.26
PerLTQA-750	51.01	66.55	58.78 (58.83 / 58.89 / 58.61)
Placebo-750	51.70	66.31	59.01 (58.92 / 59.40 / 58.70)
OME-750	52.09	66.29	59.19 (59.33 / 59.46 / 58.78)
COMET-1250	51.45	66.55	59.00 (59.78 / 58.68 / 58.54)

Table D12: Paired cluster bootstrap for end-to-end answering, OME-750 against each comparison retriever. Differences in accuracy (percentage points) per trainer seed, with 95% cluster-bootstrap intervals and two-sided p-values (2,000 replicates): the six-cell macro under the LLM judge, then the three LoCoMo cells alone under the LLM judge and under F1. The base embedder is evaluated once and is compared with each OME-750 seed.

Comparison	Seed 42	Seed 43	Seed 44
Macro, judge: PerLTQA-750	+0.46 [−0.57, +1.53], p=0.359	+0.58 [−0.46, +1.65], p=0.253	+0.48 [−0.66, +1.66], p=0.427
Macro, judge: Placebo-750	+0.49 [−0.63, +1.51], p=0.383	+0.00 [−1.08, +1.05], p=0.970	+0.24 [−0.93, +1.37], p=0.694
Macro, judge: COMET-1250	−0.22 [−1.14, +0.62], p=0.605	+1.07 [+0.16, +2.05], p=0.022	+0.18 [−0.70, +1.09], p=0.685
Macro, judge: Base	+4.62 [+3.51, +5.64], p<0.001	+4.61 [+3.53, +5.66], p<0.001	+4.16 [+2.95, +5.28], p<0.001
LoCoMo, judge: PerLTQA-750	+0.63 [−0.37, +1.63], p=0.233	+1.73 [+0.59, +2.86], p=0.004	+1.47 [+0.25, +2.64], p=0.018
LoCoMo, judge: Placebo-750	+0.69 [−0.59, +1.72], p=0.289	+0.11 [−1.14, +1.21], p=0.827	+0.71 [−0.48, +1.86], p=0.244
LoCoMo, judge: COMET-1250	+0.69 [−0.27, +1.54], p=0.163	+1.86 [+1.02, +2.70], p<0.001	+0.30 [−0.44, +1.05], p=0.430
LoCoMo, judge: Base	+8.68 [+7.46, +9.79], p<0.001	+8.55 [+7.53, +9.48], p<0.001	+8.48 [+7.33, +9.58], p<0.001
LoCoMo, F1: PerLTQA-750	+0.71 [−0.10, +1.59], p=0.100	+1.70 [+0.99, +2.38], p<0.001	+0.85 [−0.05, +1.74], p=0.064
LoCoMo, F1: Placebo-750	+0.53 [−0.15, +1.18], p=0.125	+0.24 [−0.58, +1.03], p=0.563	+0.40 [−0.48, +1.22], p=0.352
LoCoMo, F1: COMET-1250	+0.23 [−0.43, +0.88], p=0.552	+1.28 [+0.54, +2.01], p<0.001	+0.43 [−0.23, +1.09], p=0.218
LoCoMo, F1: Base	+5.57 [+4.77, +6.47], p<0.001	+5.73 [+4.86, +6.61], p<0.001	+5.22 [+4.30, +6.17], p<0.001

Retrieval sensitivity of PersonaMem answering. Answering accuracy on PersonaMem depends on retrieval much less than on LoCoMo, consistent with PersonaMem accuracy moving by under 1pp across all retrievers in Tables D10–D11 while LoCoMo moves by 7–9pp. With the base embedder, questions whose gold memories are all retrieved (Recall@10 = 1) are answered correctly 62–64pp more often than questions with none retrieved on LoCoMo, but only 20–25pp more often on PersonaMem (every gap p<0.001, cluster bootstrap), and PersonaMem questions with no gold memory retrieved are still answered correctly 49–54% of the time, against roughly 25% by chance with four options. Questions whose evidence Mem0 lost at write time, which no retriever can recover, are answered only 5.1pp less accurately than questions whose evidence was stored (60.1% against 65.3%; 95% CI [−4.48, +14.20], p=0.316; 20 persona clusters). EvoEmbedding reports the same pattern: at every retrieval budget, PersonaMem accuracy spans only 2–5 points across twelve embedders whose LoCoMo accuracy spans 27–37 points (per its appendix), and representation collapse costs about 8% on PersonaMem against over 50% on LoCoMo, which its authors attribute to the multiple-choice format (Nie et al., 2026). For this reason, we believe results for LoCoMo are more interpretable given its variation between embedders and greater dependence on information within memories, and report it as our primary end-to-end answering metric.

APPENDIX D.6 COMET PROXIES FOR AUGMENTED DATASETS

Table D13: Change in the targeted COMET proxy from the source PerLTQA query to the augmented query, by COMET dataset. Each row pairs every augmented query with the original PerLTQA query it was built from and reports the mean of that dataset's targeted proxy under each query, with the mean paired change. All proxies are computed with the same annotation procedure and formulae described in Appendix A. n is the number of eligible comparison queries; temporal mismatch is low because the majority of queries contain no temporal expressions. The memory-query form row is flat because that proxy is already at its ceiling on the selected source queries (mean 0.79, median 1.0), leaving the rewrites no headroom to increase it.

COMET dataset	Targeted proxy	n	Source mean	Augmented mean	Mean paired change
Concentration (C-250)	Concentration	250	0.687	0.973	+0.285
Over-fragmentation (O-250)	Fragmentation	250	1.000	2.000	+1.000
Memory-query form (M-250)	Form gold mismatch	245	0.788	0.779	-0.007
Entity (E-250)	Entity gold mismatch	248	0.127	0.304	+0.176
Temporal (T-250)	Temporal gold mismatch	67	0.821	0.940	+0.123

We note that all proxies are increased substantially by their targeted dataset (over-fragmentation is simply the number of ground-truth memories, which the augmented queries have two while the original queries has one) except for memory-query form mismatch (M-250), which we attribute to two reasons. First, the original source mean was already fairly high (0.788), close to complete lexical disjointness between query and memory weighted by IDF; it is difficult to further increase the proxy without completely altering the information need of the question. Second, the memory-query form proxy measures lexical overlap, which is an imperfect (but computationally tractable) proxy for what the failure mode intends to capture: whether the query and memory express the same information need in lexically-different forms. We present randomly chosen examples from every COMET dataset below to show that each augmented dataset indeed corresponds to its desired category.

APPENDIX D.7 CASE STUDY

Tables D14-D19 provide five examples per augmented dataset (each COMET category plus the placebo), sampled uniformly at random with seed 42 to avoid cherrypicking. Each row gives the source PerLTQA question, the augmented query the adapter is trained on, and the ground-truth memory that carries the answer; the augmentation changes the form of the query and never its ground-truth memory. Augmented queries are reproduced verbatim, while memory text is truncated for display. Over-fragmentation is the only dataset whose target spans two memories, as its rewrites deliberately link evidence that is fragmented across them.

APPENDIX D.7.1 CONCENTRATION

Table D14: Concentration case study.

#	Source query	Augmented query	Target memory (excerpt)
1	What is the best way to blend highlight and shadow colors for a natural look?	When blending contour colors, what did Peng Jie recommend for a natural three-dimensional effect?	Summary: Peng Jie recommended Estée Lauder double-effect makeup fixing cream to Zhang Tao On March 15, 2022, Peng Jie met his friend Zhang Tao at a community event. Peng Jie was very interes…

#	Source query	Augmented query	Target memory (excerpt)
2	Which animals did Xu Hui mention as inhabitants of the Amazon rainforest?	In her Amazon rainforest discussion, which animals did Xu Hui list?	Summary: Discussing the Amazon Rainforest Ecosystem At the academic conference on June 10, 2022, Xu Hui and her colleague Wang Ming occasionally exchanged some research ideas, including a di…
3	What additional feature does the sound system offer?	What multi-room capability did Cai Xiuying's home sound system provide?	Summary: Buy the BOSE SoundTouch 300 sound system On a sunny weekend, Cai Xiuying and her mother Cai Yanying spent a pleasant family time together. The two decided to upgrade their home ente…
4	How did Li Hua respond to Zheng Yong's suggestions for improving the community environment?	How did Li Hua respond to the community-improvement suggestions?	Summary: Participate in community activities During the community activity on October 12, Zheng Yong and Li Hua participated in the construction of the community. Together, they participated…
5	During the faith sharing meeting, who did Wang Tao engage with to discuss the relationship between faith and life goals?	Which group of people did Wang Tao discuss faith and life goals with?	Summary: Attend a faith sharing meeting On May 15, 2022, Wang Tao, an educator in Huang Xin's community, participated in a faith sharing meeting held at the Prayer Café in New York City. Thi…

APPENDIX D.7.2 OVER-FRAGMENTATION

Table D15: Over-fragmentation case study.

#	Anchor question	Augmented query	Target memory 1 (excerpt)	Target memory 2 (excerpt)
1	What did Zhou Ting inquire about when speaking to Professor Yang Fan?	What did Zhou Ting inquire about when speaking to the person who served as Zhou Ting's mentor during her time at Li Ka-shing University?	[2022-03-15 10:00] Zhou Ting: Professor Yang, I've been thinking a lot about my career path after graduation. [2022-03-15 10:00] Yang Fan: It's crucial to start planning early, Tin…	Summary: Yang Fan guided and supported Zhou Ting's career development during college. While studying at Li Ka-shing University, Zhou Ting met Professor Yang Fan and became her ment…
2	What is the relationship between Liu Liang and Chen Yu?	What is the relationship between Liu Liang and the person with whom Liu Liang has recently conducted in-depth research and communication?	[2022-05-12 10:00] Liu Liang: Chen Yu, I was thinking about getting a new smartphone but I'm not sure where to start. Could you give me some advice? [2022-05-12 10:00] Chen Yu: Of…	[2022-03-15 10:00] Liu Liang: Chen, I've been reviewing the administrative licensing issues ABC Chemical Co., Ltd. faced. It's been quite a challenging process for them. [2022-03-1…

#	Anchor question	Augmented query	Target memory 1 (excerpt)	Target memory 2 (excerpt)
3	What roles did Feng Wei and Liu Fang fulfill during the bullwhip disease awareness event?	What roles did the person with whom Wang Li collaborates closely in the 'Harmonious Sound of the City'project and Liu Fang fulfill during the bullwhip disease…	Summary: Participate in bullwhip disease awareness campaigns As one of the community environmental protection volunteers, Feng Wei and community member Liu Fang participated in the…	Summary: Participated in the 'Harmonious Sound of the City'music city planning project At a climate change seminar, Feng Wei met his girlfriend Wang Li. Wang Li is a music lover a…
4	What role did Zhang Ming play in the environmental protection case?	What role did the person who wore traditional opera costumes and facial makeup during the performance play in the environmental protection case?	[2022-05-10 10:00] Zhao Li: Ming, I really appreciate your help with gathering all that data on the refinery emissions. [2022-05-10 10:00] Zhang Ming: It's no problem, Lili. My bac…	[2023-10-01 10:00] Zhao Li: Ming, I heard your performance at the Cultural Exchange Festival was a huge success! [2023-10-01 10:00] Zhang Ming: Yes, Lili! It was incredible. We pre…
5	Is Li Jiacheng inclined to purchase Haier products again in the future?	Is the person who is accompanying Liang Xin inclined to purchase Haier products again in the future?	Summary: Haier home appliance case In terms of quality inspection in the shopping and consumer fields, we take Haier home appliances as an example to discuss. As a consumer, neighb…	[2023-10-21 10:00] Liang Xin: It's such a beautiful afternoon to be sitting outside, don't you think, Jiacheng? [2023-10-21 10:00] Li Jiacheng: Absolutely, Xinxin. The weather is p…

Appendix D.7.3 Memory-query form mismatch

Table D16: Memory-query form mismatch case study.

#	Source query	Augmented query	Target memory (excerpt)
1	What qualities did Qin Meng and Wang Xuan demonstrate during the furniture repair process?	What approach did Qin Meng take to a household problem outside her expertise?	Summary: Participate in bedroom furniture repair together Qin Meng participated in the repair of bedroom furniture with her boyfriend Wang Xuan. They found bedside table drawers stuck, cupbo…
2	What did Yang Ming's guidance help establish in his relationship with Li Hua?	What creative priority guided Yang Ming's mentorship of Li Hua?	Summary: Guiding Li Hua on his painting techniques and art theory During Li Hua's artistic career, he was fortunate enough to have Yang Ming as his mentor. Yang Ming is a well-known artist w…

#	Source query	Augmented query	Target memory (excerpt)
3	What details did Liu Hai include in his letter of support for Li Hua?	What support role did Liu Hai take in a workplace-discrimination complaint?	Summary: bangs support Liu Hai learned of the situation after Li Hua filed a formal complaint with the company's human resources department. As a college classmate, he knew Li Hua's talents…
4	Did the assistant agree to help Cao Lili with translating the English composition?	What learning approach did Cao Lili take in shared study sessions?	[2023-10-25 10:00] Cao Lili: It's a bit chilly today, isn't it, Liu Lei? A perfect afternoon for focusing on our studies. [2023-10-25 10:00] Liu Lei: Absolutely, Xiaoli. The cold seems to ma…
5	What skills do Yang Wei and Liu Lei bring to their environmental protection projects?	What design strategy guided Yang Wei's sustainable urban work?	[2022-05-15 10:00] Yang Wei: I'm really pleased with how our project turned out, Liu Lei. The integration of greening systems with urban planning was a massive challenge. [2022-05-15 10:00]…

Appendix D.7.4 Entity mismatch

Table D17: Entity mismatch case study.

#	Source query	Augmented query	Target memory (excerpt)
1	What criteria do Qin Meng and Chen Ming use to select conditions for analyzing public opinion?	What criteria do Qin Meng and the colleague who analyzed Google Trends data with her use to select conditions for analyzing public opinion?	Summary: Qin Meng and Chen Ming cooperate to analyze Trends platform data In the case of Google Trends platform public opinion monitoring, Qin Meng collaborated with her colleague Chen Ming…
2	Why is Cao Ming particularly fond of the Chinese Film New Power Award category?	Why is Cao Lili's movie-loving brother who took her to the 1993 Shanghai International Film Festival particularly fond of the Chinese Film New Power Award category?	Summary: Attend Shanghai International Film Festival The 1993 Shanghai International Film Festival was the first time Cao Ming participated in this event. As Cao Lili's brother, Cao Ming has…
3	What roles do Sun Xiaoming and Liu Xiang play in organizing environmental protection activities?	What roles do Sun Xiaoming and the co-organizer who shared ecological-protection photographs with him play in the environmental activities?	Summary: Participate in environmental activities Sun Xiaoming and Liu Xiang are community members who participated in organizing an environmental protection event to improve the ecological e…
4	What actions did the doctor take upon arriving at Cai Xiuying's residence?	What did the community health-service clinician whom Zhang Peng contacted do upon reaching Cai Xiuying's home?	Summary: Help Cai Xiuying On this day, Cai Xiuying was preparing to attend an important meeting of an international research institution. However, she suddenly felt unwell and developed symp…

#	Source query	Augmented query	Target memory (excerpt)
5	What activities do Cai Xiuying and Chen Ming enjoy during their travels?	What travel activities do the international-politics researcher and the romantic partner who reads her papers and models for her photographs enjoy?	Summary: Experience wonderful times together Cai Xiuying and Chen Ming are a romantic couple who support each other and experience many wonderful times together. During Cai Xiuying's busy wo…

APPENDIX D.7.5 TEMPORAL MISMATCH

Table D18: Temporal mismatch case study.

#	Source query	Augmented query	Target memory (excerpt)
1	In what ways does Yang Fan assist Zhou Ting in expanding her professional knowledge?	Earlier in her career, in what ways did Yang Fan assist Zhou Ting in expanding her professional knowledge?	Summary: The important influence of mentors on Zhou Ting's career development Zhou Ting's mentor during college, Yang Fan, had an important influence on her career development. Yang Fan is a…
2	How did Xu Hui feel on the evening of May 15, 2022?	That evening in spring 2022, how did Xu Hui feel?	Summary: Xu Hui shares her academic passion with Zhang Yu On a warm evening on May 15, 2022, Xu Hui and Zhang Yu were sitting in the study room at home, with a glass of champagne and a book…
3	What roles did Zhou Ting and Zhang Tao have in the recent volunteer activity?	Lately, what roles did Zhou Ting and Zhang Tao have in the volunteer activity?	Summary: volunteer activity Zhou Ting is one of the organizers of community activities. She and Zhang Tao are community members and often participate in community activities. Recently, they…
4	What insights and experiences did Xu Jia share during the online lecture on film production?	At the online lecture that spring, what film-production experience and techniques did Xu Jia share?	Summary: Zhang Li attended an online lecture on film production on TalkWorld On May 15, 2022, Zhang Li, as a film production student, participated in an online lecture on film production on…
5	What additional steps have Peng Jie and Wang Li taken to enhance family relationships?	By autumn 2018, what additional steps had Wang Li's family taken to strengthen their relationships?	[2018-09-15 10:00] Peng Jie: I heard you and Li Ming are taking a communication training course. How is it going? [2018-09-15 10:00] Wang Li: Yes, we started it a few weeks ago. It's been in…

APPENDIX D.7.6 PLACEBO (GENERIC AUGMENTATION)

Table D19: Placebo (generic augmentation) case study.

#	Source query	Augmented query	Target memory (excerpt)
1	Is Kong Tingting familiar with the metamaterial shell technology developed by Stanford University?	Does Kong Tingting know about the metamaterial shell technology developed by Stanford University?	[2021-06-12 10:00] Kong Tingting: Xiao Ming, I've found something fascinating in the library today. It's a research book on metamaterials! [2021-06-12 10:00] Xiao Ming: Really? What are meta...
2	How are Liu Liang and Wang Li related?	What is the relationship between Liu Liang and Wang Li?	Supporting Characters: Wang Li Relationship: girlfriend Description: Liu Liang's girlfriend is an environmental volunteer and enthusiastic about public welfare.
3	What were the most significant achievements of the Aztec Empire?	Which accomplishments of the Aztec Empire were the most significant?	Summary: Zhou Ting and Li Ming's family gathering Zhou Ting attended a family gathering with her younger brother Li Ming. The gathering was to celebrate Li Ming's important victory at the law firm. At the party, Zhou Ting shared the case of the Aztec Empire with Li Ming and other family members, mentioning its establishment and rule and its important political, military, religious and economic achievements...
4	What is Huang Ming's reaction to learning about Toon Boom Animation Studio?	How does Huang Ming react to finding out about Toon Boom Animation Studio?	Summary: Huang Ming's interest in Toon Boom Animation Studio On May 10, 2022, Huang Xin and his father Huang Ming were sitting in the living room of their home. Huang Ming has always been ve...
5	What are Wang Tao's primary responsibilities during the 'Colorful Art Journey'exhibition?	During the 'Colorful Art Journey'exhibition, what are Wang Tao's main responsibilities?	[2022-05-15 10:00] Li Hua: Wang Tao, have you finalized the gallery booking for the 'Colorful Art Journey'exhibition? [2022-05-15 10:00] Wang Tao: Yes, Xiaohua. I confirmed the venue at the...

APPENDIX D.8 ADAPTATION GAINS ON HIGH-MISMATCH QUESTIONS

Table D20: Gain in Recall@10 over the base embedder below and above the median of each adapter's own COMET proxy. Each row is stratified on the question-level percentile of the gold-side proxy corresponding to that adapter, split at the median, and reports the adapter's mean Recall@10 minus the base embedder's within that stratum (percentage points, three-seed means of Qwen3-Embedding-0.6B adapters). Strata are formed within the two cells with a matching frozen RQ1 proxy artifact (RAG and dense Mem0 on LoCoMo) and the two cells are averaged; questions for which a proxy is undefined are excluded. Bold marks the upper stratum where its gain exceeds the lower stratum's.

COMET adapter	Proxy	Gain, below median	Gain, above median
Concentration	Concentration	+8.97	**+10.69**
Over-fragmentation	Fragmentation	+11.61	**+13.72**
Memory-query form	Form gold mismatch	+13.02	+12.36
Entity	Entity gold mismatch	+9.63	**+13.60**
Temporal	Temporal gold mismatch	+14.00	+13.38

APPENDIX E HUMAN AUDIT RESULTS

Here we audit all LLM-involved annotation steps within our experimental pipeline, including:

- Entity, time-expression and relevant-span annotations for queries and memories in each system; for relevant spans, we audit whether they truly cover all relevant spans in the memory to the query (coverage) and whether all are relevant (exclusivity).
- Ground-truth memory selection for Mem0 only; we audit whether the LLM's judgment on sufficiency is correct (i.e. whether the candidate memories have any sufficient subset of ground-truth memories for the query), whether its selected subset is sufficient, whether it is minimal and whether it is unique.

The audit sampled 20 cases per benchmark/system stratum (with the annotations being embedder- and k-agnostic), with a balanced set of cases across all question categories on each benchmark (5 per the four categories in LoCoMo, 3 per category with one category receiving 2 across the seven categories in PersonaMem). The human reviewer assessed the query annotations, the displayed memory annotations, and the listed relevant spans. We additionally used a coding agent, GPT-6 Astra hosted through the Codex wrapper, to independently conduct the same audits; the human and Astra reviews share 115 of the 120 sampled cases, and Table E1 reports the results on those jointly reviewed cases. Positive rates are the proportion judged valid or relevant.

Table E1: Human and Astra audits for COMET annotations. The audit sampled 120 cases; the human and Astra reviews share 115 of them, and all three columns are computed on those 115 jointly reviewed cases.

Audit dimension	Human positive rate	Astra positive rate	Paired raw agreement
Query entities valid	92/115 (80.0%)	104/115 (90.4%)	95/115 (82.6%)
Query times valid	112/115 (97.4%)	115/115 (100.0%)	112/115 (97.4%)
Memory entities valid	107/115 (93.0%)	101/115 (87.8%)	97/115 (84.3%)
Memory times valid	107/115 (93.0%)	115/115 (100.0%)	107/115 (93.0%)
Relevant-span coverage	111/115 (96.5%)	112/115 (97.4%)	108/115 (93.9%)
Relevant-span exclusivity	107/115 (93.0%)	115/115 (100.0%)	107/115 (93.0%)

The audit indicates that the generated annotations and relevant spans are generally reliable. The lowest human rate is for query entities (80.0%), followed by memory entities, memory times and relevant-span exclusivity (93.0%). Query times and relevant-span coverage are above 96% in the human audit. Astra's independent judgments show the same pattern of broadly high validity.

APPENDIX E.1 MEM0 GROUND-TRUTH MEMORY-SELECTION AUDIT

The separate Mem0 audit sampled 50 cases per benchmark. The confusion matrix compares the auditor's judgment that some eligible memory subset supports the benchmark answer with the pipeline's recorded gold-selection status. Sufficiency, minimality, and uniqueness are reported conditionally on cases that the pipeline marked sufficient, i.e. cases with a selected ground-truth set to assess (the true-positive and false-positive cells together).

Table E2: Confusion matrix for Mem0 ground-truth annotations. TP = true positive (sufficient marked sufficient); FP = false positive (insufficient marked sufficient); TN = true negative; FN = false negative.

Benchmark	Reviewer	TP	FP	TN	FN
LoCoMo	Human	38	0	8	4
LoCoMo	Astra	42	6	2	0
PersonaMem	Human	35	3	9	3
PersonaMem	Astra	38	0	12	0

Table E3: Human and Astra audits for Mem0 ground-truth annotations. Eligible cases are those the pipeline marked sufficient (TP + FP in Table E2), and each conditional rate is reported over that eligible set as the denominator.

Benchmark	Reviewer	Eligible cases	Selected set sufficient	Selected set minimal	Selected set unique
LoCoMo	Human	38	37/38 (97.4%)	36/38 (94.7%)	34/38 (89.5%)
LoCoMo	Astra	48	42/48 (87.5%)	42/48 (87.5%)	37/48 (77.1%)
PersonaMem	Human	38	35/38 (92.1%)	34/38 (89.5%)	31/38 (81.6%)
PersonaMem	Astra	38	38/38 (100.0%)	38/38 (100.0%)	38/38 (100.0%)

These percentages are calculated only from cases where the pipeline selected a ground-truth set (i.e. did not record a write failure) in each benchmark. Among those cases, selected-set sufficiency was 37/38 (97.4%) in LoCoMo and 35/38 (92.1%) in PersonaMem; minimality was 36/38 (94.7%) and 34/38 (89.5%), respectively; uniqueness was 34/38 (89.5%) and 31/38 (81.6%). Astra's corresponding conditional rates are shown alongside the human results.

APPENDIX E.2 END-TO-END JUDGE ACCURACY

The end-to-end accuracy labels are produced by an LLM judge, so we audit the judge itself. The sample is 90 completed LoCoMo judgments, LoCoMo being the only judged benchmark because PersonaMem answers are scored deterministically by multiple-choice exact match. Cases were drawn as 15 per memory system crossed with 15 of the judge's CORRECT labels and 15 of its WRONG labels, pooled across adapter conditions, and labelled blind to the judge's verdict; the sampling seeds are recorded in the run's manifest. Human–judge agreement is 91.6% re-weighted to the population mix of verdicts (80 of 90 as sampled, Cohen's $\kappa = 0.78$). Within the judge's CORRECT calls the auditor agreed on 45 of 45, and within its WRONG calls on 35 of 45. Every disagreement is a case in which the judge rejected an answer that carries part of the gold answer, so the judge is applying its containment rubric strictly and does not accept incorrect answers; its errors fall on the rejection side, which makes the reported accuracies conservative rather than inflated.

Agreement is also comparable across systems, at 27 of 30 for RAG, 27 of 30 for Mem0 and 26 of 30 for A-MEM, a spread consistent with chance, so the judge's error does not track the memory system. An independent automated re-review (Astra, run blind under the same rubric) reproduced the judge's labels on all 90 cases (90/90, Cohen's $\kappa = 1.0$); its agreement with the human auditor is therefore also 88.9%, matching the judge's.

Table E4: Human audit of the end-to-end answer judge. Agreement between the auditor and the judge by the judge's verdict, over 90 completed LoCoMo judgments sampled as 15 per memory system crossed with 15 CORRECT and 15 WRONG labels. Agreement is 91.6% when re-weighted to the population mix of verdicts.

Judge verdict	Cases	Human agreement	Astra agreement
CORRECT	45	45 (100.0%)	45 (100.0%)
WRONG	45	35 (77.8%)	45 (100.0%)
All	90	80 (88.9%)	90 (100.0%)

APPENDIX F EXACT PROMPTS

The prompts below are used to create the COMET annotations and the COMET-guided adaptation datasets, and for end-to-end answering and judging (Prompts B7 and B8). They are indexed Prompt [Letter][Number], where [Letter] is the Appendix section they correspond to and [Number] is their order. These prompts are reproduced verbatim from the code used to generate the experimental artifacts, including system prompts, user prompts (e.g. JSON payloads), and additional addendums based on benchmark or system; however, the code released alongside the paper should be treated as the ultimate source of truth for the final prompts used.

APPENDIX F.1 PROMPTS FOR APPENDIX A: THE COMET FAILURE PATTERNS

APPENDIX F.1.1 PROMPT A1 — MEM0 GROUND-TRUTH MEMORY SELECTION

System prompt

```
## TASK

You are an expert linguistic annotator. You will be given a query, its benchmark answer,
and candidate memories. Select the minimal set of candidate memories required to support
the benchmark answer under the rules below.

## RULES
- Use only the supplied evidence and its displayed metadata. The benchmark answer is the
 target to support, not evidence that may be used by itself.
- A sufficient selection must jointly support the benchmark answer, use the fewest memories
 possible, and contain no removable memory.
- If all supplied evidence together does not support the benchmark answer, return "insufficient".
- If more than one minimal selection is valid, choose the one whose candidate positions form the
 lexicographically smallest sequence. List selected IDs in their original candidate order.

## REQUIREMENTS
- Reference dates are part of the memories. When the query contains a temporal expression,
 also use the query reference date.
- Copy IDs exactly. Do not output memory text or invent IDs.
- Counting and list questions require every contributing memory needed for the benchmark answer.

## OUTPUT SCHEMA
- Output exactly one strict JSON object and no Markdown:
{
 "status": "sufficient" or "insufficient",
 "selected_item_ids": ["memory_id", "..."],
 "reason": "brief explanation"
}

For "insufficient", selected_item_ids must be an empty list.
```

PersonaMem addendum

```
## PERSONAMEM CONTEXT
- The STATIC PERSONA is always supplied to the downstream answerer. Treat it as available
 evidence, but never copy it into selected_item_ids.
- Select the smallest memory set that, together with the query and STATIC PERSONA, supports
 the benchmark answer. A memory does not need to repeat a fact already stated in the persona.
- If the query and STATIC PERSONA support the answer without any candidate memory, return
 "no_memory_required" with an empty selected_item_ids list.
- Return "insufficient" only when the STATIC PERSONA and all candidate memories together
 still do not support the benchmark answer.

## OUTPUT SCHEMA
- Output exactly one strict JSON object and no Markdown:
{
 "status": "sufficient", "insufficient", or "no_memory_required",
```

```
 "selected_item_ids": ["memory_id", "..."],
 "reason": "brief explanation"
}

For "insufficient" and "no_memory_required", selected_item_ids must be an empty list.
```

User prompt

```
QUERY:
<query>

BENCHMARK ANSWER:
<benchmark answer>

QUERY REFERENCE DATE:
<query reference date or None>

STATIC PERSONA (always available; do not select it):
<static persona; PersonaMem only>

CANDIDATE MEMORIES:

MEMORY ID: <candidate item_id>
SOURCE SESSION: <candidate source_session_id or None>
SOURCE DIALOGUE IDS: <comma-separated candidate source_dialogue_ids>
REFERENCE DATE: <candidate reference_date or None>
TEXT:
<candidate text>

[The candidate-memory block above is repeated once for each candidate in the deterministic order
supplied by the code; the STATIC PERSONA block is omitted outside PersonaMem.]
```

APPENDIX F.1.2 PROMPT A2 — ENTITY AND TIME EXPRESSION ANNOTATION

System prompt

```
## TASK

You are an expert text annotator. Your task is to annotate the following text, extracting two
important pieces of information: the ENTITIES within it, and the TIME EXPRESSIONS within it.

### ENTITIES

An entity is a specific referent explicitly mentioned in the text.

Entities include specific:

- People
- Organizations
- Social groups
- Places and facilities
- Physical objects
- Events and meetings
- Documents, projects, or products

A definite description can be an entity even when it is not a proper name, such as:

- the company
- her manager
- the support group
- that meeting

Do not extract as entities:

- Generic classes
- Properties or qualities
- Numerical values
- Actions or relations
- Times or dates
- Noun phrases used only as nonreferential categories

### SERIALIZATION BOILERPLATE

Some A-MEM inputs are wrapped in labels such as Talk start time:, Memory content:,
Memory context:, Memory keywords:, and Memory tags:. These labels are formatting
metadata, not entities. Likewise, in a wrapper such as `Speaker John says:`, annotate
John when it is a referent, but do not annotate Speaker, speaker, the speaker,
listener, the listener, or Talk merely because those role labels occur in the
serialization. Only annotate a role expression when the surrounding text clearly uses
it as a specific referent rather than as a wrapper label.

### ENTITY REQUIREMENTS

You are required to:
- Identify all unique entities in the text.
- For each entity, identify all surface forms in the text that co-refer to the entity.

For an entity, a phrase should only be counted as a surface-form for that entity if you are
```

```
ABSOLUTELY CONFIDENT that the given text alone supports that association.

Every surface form must be copied exactly from the text.

Do not count the following expressions requesting for an unknown entity/answer as an entity in
themselves:

- who
- whom
- what
- which person
- which company
- where

These are answer variables, not co-referring entity mentions for this task.

Continue to annotate ordinary entity mentions elsewhere in the same question.

Do NOT output a pronoun as an entity mention or surface form. This includes personal, possessive,
reflexive, and standalone demonstrative pronouns such as "I", "me", "you", "he", "her", "it", "we",
"them", "mine", "his", "hers", "ourselves", "this", and "that".

If a pronoun co-refers with a named or descriptive entity, omit the pronoun rather than adding it to
that entity's surface-form group.

Examples:
"Priya said she attended the meeting."
Entities: [["Priya"], ["the meeting"]]
Do not include "she".

"Her manager contacted the company."
Entities: [["Her manager"], ["the company"]]
Do not separately include "Her".

"The company said it would respond."
Entities: [["The company"]]
Do not include "it".

A complete definite description may contain a possessive or demonstrative determiner when it is
needed to identify a different referent. For example, "her manager" and "that meeting" may be
entities, but do not separately output "her" or "that".

Use the smallest complete contiguous expression that identifies the entity.

Examples of valid boundaries:

- Priya
- the community center
- her former manager

For proper names:

- Include the complete name.
- Exclude external titles, possessive endings, appositives, and surrounding descriptions.

For descriptions:

- Include determiners, possessives, and restrictive modifiers needed to identify the referent.

Nested entity mentions are allowed only when the complete expressions refer to different entities.
Do not separately annotate a pronoun contained within a larger description.

For a coordinated group, use the complete coordination. Do not pair a plural group expression with
only one member of the group.

Do not emit both a full name and a shorter subspan for the same mention.

### TIME

Annotate only an explicit temporal expression that asserts or constrains a time, date, duration,
frequency, or interval. Annotate only the minimal span required to uniquely identify that time
expression and carries all the information related to it.

Examples:

- yesterday
- last week
- August 2023
- for three years
- two days ago
- before the meeting
- every Monday

### TIME REQUIREMENTS

You are required to:
- Identify all temporal expressions in the text.
- Resolve each expression to a specific absolute time interval or timestamp only when this is
defensible from the text and reference date. Otherwise, use null for both interval endpoints.
- Copy each temporal expression exactly from the text.

Do not annotate an expression that merely requests temporal information:

- when
- what time
```

```
- what date
- how long
- how long ago

A question beginning with When is not in itself a time expression, unless it also contains a
separate explicit temporal expression that specifies some specific period of time.

### OUTPUT SCHEMA

Output exactly one strict JSON object and no Markdown:

{
"entities": [["surface form 1", "co-referring surface form 2"], ["surface form of another entity"]],
"times": [{
"text": "exact temporal expression",
"interval_start": "YYYY-MM-DD or null",
"interval_end": "YYYY-MM-DD or null"
}]
}

Interval endpoints are inclusive. They must either both be null or both be valid ISO-8601 dates. A
single date or moment can have interval_start = interval_end.
```

User prompt

```
{
 "reference_date": "<reference date or null>",
 "text": "<text>"
}
```

APPENDIX F.1.3 PROMPT A3 — RELEVANT-SPAN ANNOTATION

System prompt

```
## TASK

You are an expert text annotator. You will be given one query and one memory.
Return every non-overlapping span from the memory whose information contributes
positively to recognizing that memory as semantically relevant to the query.
Judge this memory independently; no benchmark answer, retrieval result, or other
memory is available.

## RULES

- Every span must be an exact contiguous substring of the supplied memory.
- Conceptually divide the memory into the smallest non-overlapping,
 independently interpretable phrases or clauses.
- For each phrase or clause, imagine removing the information it expresses while
 leaving the rest of the memory grammatical and coherent. Return it when doing
 so would make a competent reader meaningfully less able to recognize why the
 memory is relevant to the query.
- A span need not answer the query directly. Include identifiable evidence,
 events, entities, relationships, temporal information, subject matter, and
 local context that establish or explain relevance.
- Judge informational relevance, not similarity of wording. Treat paraphrased
 and differently worded evidence the same as lexically matching evidence.
- Preserve enough local context to make every span independently interpretable,
 including any necessary subject, predicate, object, negation, quantity,
 temporal qualifier, or antecedent. Do not return an isolated answer token,
 entity, or date when its relationship to the query would be unclear.
- Return spans in their order in the memory. Do not return overlapping spans.
- Include separate repeated occurrences when each independently supports
 relevance. If identical text occurs more than once but only one occurrence is
 relevant, enlarge the returned phrase or clause just enough to identify that
 occurrence uniquely.
- If removing a phrase or clause would leave relevance unchanged or increase it,
 do not return it.
- Use reference dates only to interpret relative temporal language. A displayed
 date is metadata, not memory evidence, and must not be returned unless those
 characters occur in the memory text.
- If no span satisfies these rules, return an empty list.

## OUTPUT SCHEMA

Return exactly one strict JSON object and no Markdown:
{"spans": ["first exact substring", "second exact substring"]}
```

User prompt

```
{
 "memory": "<memory text>",
 "memory_reference_date": "<memory reference date or null>",
 "query": "<query text>",
 "query_reference_date": "<query reference date or null>"
}
```

APPENDIX F.2 PROMPTS FOR APPENDIX B: DIAGNOSIS-GUIDED ADAPTATION

APPENDIX F.2.1 PROMPT B1 — CONCENTRATION COMET AUGMENTATION

System prompt The system message is the default Codex agent instruction supplied by the Codex runtime; it is implicit here and is not stored in this repository.

User prompt

```
Construct a set of exactly 250 concentration questions from the supplied PerLTQA training data.

Do only question construction, provenance validation, and artifact writing. Perform the semantic
judgments yourself in this Codex task. Do not call an external language model or API. Do not train
or evaluate models, construct or mine negatives, inspect retrieval rankings, proxy scores,
downstream data, downstream results, previous adapter performance, or existing generated question
artifacts.

Use only PerLTQA training questions, answers, labelled support annotations, labelled gold-memory
IDs, and the corresponding complete rendered memories. Use the existing frozen training-character
split without changing it.

For every accepted record:

1. Select one existing source question and exactly one labelled gold parent memory.
2. Preserve the source answer as the constructed answer.
3. Preserve the complete memory. Do not split, window, shorten, summarize, concatenate, pad, or
rewrite it.
4. Rewrite only the query into one direct, natural question about the same locally supported fact.
5. Use exactly one positive parent per question. The downstream preparation stage will use ordinary
complete parent-memory negatives; do not introduce any positive-versus-negative passage-form
difference.

The intended mechanism is:

A semantically precise but lexically non-identifying question whose relevance to the complete
positive memory is carried primarily by one small, answer-bearing region inside a long and
internally competitive parent memory.

More simply:

Be specific about the requested answer slot and relation, but use only the minimum broad context
needed to identify the information need.

The goal is not to write an extremely specific question about a tiny fact. Exact project names,
dates, measurements, product models, venues, titles, and multiple proper names can fingerprint the
correct parent and eliminate the retrieval-level concentration problem.

The goal is also not to write an ambiguous question such as "Which brand was used?" The requested
answer type and relation must remain explicit.

Use this construction procedure:

1. Identify the exact answer and one local sentence or dialogue turn that directly supports it.
2. Temporarily construct the question from that local relation alone.
3. Add only the minimum broad contextual anchor required to make the question natural,
self-contained, and unambiguous.
4. Restore the complete parent and verify that almost all substantive query-relevant information
remains concentrated in the support unit.
5. Reject the candidate if disambiguation requires stacking rare, episode-specific identifiers into
the query.

For example, do not write:

"In the Park Hyatt Apartments refrigeration system, which equipment brand supplied the chilled-water
unit rated at 120 kW?"

That query fingerprints the parent through a project name, technical system, equipment description,
and numerical specification.

A better construction is:

"For the apartment cooling project, which manufacturer provided equipment?"

This keeps the requested slot and relation precise while using one broad contextual anchor.

Every accepted memory must satisfy all of the following:

- the answer occurs in one localized natural-body sentence or dialogue turn;
- removing that support unit makes the answer unsupported;
- the complete memory contains substantial non-answer material before and after the support;
- there are at least three substantive units before and three after it;
- the exact support span occupies no more than 5% of the natural parent body;
- the complete parent contains at least three plausible competing facts of the same broad answer
type outside the support unit;
- the answer is not repeated or independently supported elsewhere;
- the support is not supplied by Summary, Theme, title, Creation Time, profile fields, headings, or
serialization metadata;
- no multi-memory reasoning is required.

The question must:
```

```
- state the requested answer type or semantic slot;
- state the relevant relation directly;
- contain at most one broad contextual anchor beyond an ordinary subject name;
- use ordinary entity names directly rather than creating entity mismatch;
- avoid temporal conversion, implicit trait inference, unusual syntax, or deliberate lexical
obfuscation;
- remain easy to answer once the correct complete memory is available.

Reject a candidate if the question unnecessarily contains any combination of:

- an exact date or numerical specification;
- a product or model number;
- a complete event, project, paper, or exhibition title;
- a rare venue or organization name;
- multiple named participants;
- an uncommon phrase copied from a summary or theme;
- several details that jointly identify the complete episode.

If such details are necessary to make the question unambiguous, reject the candidate and choose
another case rather than retaining the fingerprint.

Apply two further tests.

First, the localization test:

Apart from the ordinary subject and one broad contextual anchor, the substantive meaning of the
query must map primarily to the answer-bearing support unit. Reject the question if several of its
identifying details come from unrelated parts of the complete parent.

Second, the ambiguity test:

Without using embeddings or retrieval rankings, inspect the other eligible complete memories for the
same character only to determine whether another memory could plausibly answer the constructed
question differently. Reject an ambiguous question. This is rejection-only semantic validation; do
not choose, record, or optimize against particular negatives.

Use 250 distinct source-question IDs and 250 distinct character-qualified positive parent-memory
IDs. Do not accept separate renderings of the same underlying episode as independent positives. Do
not impose equal character, answer-type, or memory-category quotas, but include both dialogue and
event memories when valid examples exist.

Before accepting each record, verify:

1. The source answer is directly supported by one local natural-body unit.
2. The selected memory is an unchanged labelled gold parent.
3. The answer-bearing signal is small relative to the complete parent.
4. At least three same-type competing facts occur outside the support.
5. Deleting the support removes the answer.
6. The answer is not repeated elsewhere in the parent.
7. The query asks for a precise answer type and relation.
8. The query contains only the minimum broad context needed for interpretation.
9. The query does not fingerprint the parent using stacked rare identifiers.
10. The question is unambiguous within the character's eligible memory store.
11. No other COMET mechanism is required to answer it.
12. The wording is one natural request a real user might ask.

Order eligible source-question and gold-memory pairs deterministically by the SHA-256 hash of:

 concentration|<source_question_id>|<gold_memory_id>

Review them in that fixed order and freeze the first 250 satisfying every rule. If the complete
training population cannot supply 250 valid records, stop and report the shortfall rather than
relaxing the rules.

Write one JSON object per line containing:

- question_id
- source_question_id
- character_id
- source_question
- source_answer
- constructed_question
- constructed_answer
- gold_memory_ids
- gold_memories
- memory_category
- underlying_episode_id
- support_spans
- support_sentence_or_turn
- support_character_count
- natural_body_character_count
- support_fraction
- substantive_units_before
- substantive_units_after
- answer_type
- target_relation
- contextual_anchor
- competing_facts
- omitted_episode_identifiers
- localization_check
- ambiguity_check
- validation_notes

Use stable sequential IDs:
```

```
concentration::0000
...
concentration::0249

gold_memory_ids and gold_memories must each contain exactly one item. competing_facts must quote at
least three exact competing clauses or spans. omitted_episode_identifiers must record distinctive
details deliberately excluded from the query; it may be empty when the memory contains none.

Write incrementally so accepted work survives interruption. Freeze the output in a new artifact
without overwriting any existing file.

The manifest must record the complete prompt, input and output SHA-256 hashes, split and ordering
provenance, candidate and rejection counts, concise rejection reasons, record count, positive-parent
and underlying-episode reuse, support-position, parent-length, support-fraction, competing-fact,
query-length, memory-category, character, and answer-type distributions, fingerprint flags, and
validation results.

Compute descriptive distributions only after semantic acceptance. Do not alter the frozen set using
retrieval difficulty or downstream information. Do not proceed to negative mining, training, or
evaluation.
```

APPENDIX F.2.2 PROMPT B2 — OVER-FRAGMENTATION COMET AUGMENTATION

System prompt The system message is the default Codex agent instruction supplied by the Codex runtime; it is implicit here and is not stored in this repository.

User prompt

```
Construct a new frozen set of 250 multi-memory questions from the existing PerLTQA training split.

Do only question construction. Do not train models, evaluate models, mine negatives, inspect COMET
proxy scores, inspect retrieval results, or use LoCoMo, PersonaMem, development data, or test data.
Do not modify existing questions, artifacts, or code. Write the new dataset to a separate,
non-overwriting JSONL file.

For each example:

1. Choose one existing PerLTQA training question as the anchor.
2. Choose one or more additional questions about the same character whose labelled gold parent
memories are distinct from the anchor memory.
3. Minimally rewrite the anchor question so that:
 - its original answer remains the answer;
 - its interrogative form, principal predicate, tense, vocabulary, and structure remain as close
to the original as possible;
 - answering it now requires evidence from every selected memory.
4. Usually accomplish this by replacing a directly named person, event, place, time, or relationship
in the anchor with a natural description that must be resolved using the additional memories.
5. Use complete, unmodified ordinary parent memories as positives. Prefer dialogue and
narrative-event memories. Reject short profile fields, gender labels, social-relationship records,
or other metadata records when they provide only a trivial lookup rather than substantive evidence.
Do not impose a memory-type quota.
6. Do not combine an event memory with a dialogue memory that merely describes the same episode.
Every memory must contribute a distinct necessary fact.
7. Use the smallest necessary number of memories. Every question must require at least two, but do
not force a 2/3/4 distribution or any maximum.
8. Reject the example if:
 - one memory alone answers the complete question;
 - removing any selected memory leaves the complete answer obtainable;
 - an added memory is merely helpful, descriptive, or redundant;
 - the question introduces an unsupported identity, chronology, relationship, or causal claim;
 - the result is unnatural;
 - the result is several questions joined together.

Do not concatenate source questions. Do not use multiple question marks, lists, numbered parts, or
wrappers such as "Answer all parts." Produce one natural information need.

Copy all selected parent-memory texts exactly. Never split, summarize, shorten, combine, clean, or
paraphrase them.

Write one JSONL record per accepted example:

{
"question\_id": "anchored-fragmentation::0000",
"character\_id": "...",
"anchor\_question\_id": "...",
"anchor\_question": "...",
"rewritten\_question": "...",
"answer": "...",
"source\_question\_ids": ["...", "..."],
"gold\_memory\_ids": ["...", "..."],
"gold\_memories": [
"exact complete parent-memory text",
"exact complete parent-memory text"
],
"memory\_roles": [
"fact this memory necessarily contributes",
"fact this memory necessarily contributes"
]
}
```

```
The lengths of source\_question\_ids, gold\_memory\_ids, gold\_memories, and memory\_roles must be
equal.

Before accepting an example, test each selected memory by mentally removing it. Accept the example
only if removing any one memory makes the complete answer unsupported or the question's referent
impossible to resolve.

Prefer not to reuse source questions or gold memories. Do not impose character quotas. Select
examples solely because they support a natural anchored rewrite.

Write accepted records incrementally so progress survives interruption. At completion, write a
manifest containing the input and output file hashes, accepted count, rejected count,
memory-cardinality distribution, character distribution, source-question reuse count, gold-memory
reuse count, and these construction instructions verbatim.

If 250 valid examples cannot be found without relaxing these rules, stop and report the number
obtained. Do not lower the acceptance standard.
```

APPENDIX F.2.3 PROMPT B3 — MEMORY-QUERY FORM-MISMATCH COMET AUGMENTATION

System prompt The system message is the default Codex agent instruction supplied by the Codex runtime; it is implicit here and is not stored in this repository.

User prompt

```
Construct a frozen set of exactly 250 form-mismatch questions from the
existing PerLTQA training data.

Do only question construction, provenance validation, and artifact writing.
Perform the semantic judgments yourself in this Codex task. Do not call an
external language model or API. Do not write a general question generator,
construct or mine negatives, modify existing code, data, memories, or
annotations, train or evaluate a model, or inspect retrieval rankings, proxy
scores, retrieval difficulty, downstream data, downstream results, previous
adapter performance, or existing generated-question artifacts.

Inspect only the frozen PerLTQA training questions and answers, their labelled
gold-memory IDs and support annotations, the corresponding complete rendered
memories, and repository code needed to read and validate those inputs.

The scientific objective is to construct queries that ask for an abstract,
implicit, relational, or high-level interpretation while the positive memory
expresses the answer through concrete actions, statements, events, outcomes, or
behaviour.

The query must not repeat the concrete episode description that identifies the
positive memory. The retriever should have to align the abstract information
need with the concrete evidence, rather than retrieve the parent using copied
names, actions, objects, titles, dates, or other episode-specific details.

For every accepted example:

1. Select one existing PerLTQA source question and exactly one labelled gold
 parent memory containing concrete evidence for an implicit interpretation.
2. Preserve the complete selected memory byte-for-byte. Do not rewrite, split,
 window, summarize, concatenate, shorten, pad, or otherwise alter it.
3. Rewrite only the query and construct one supported abstract answer.
4. Ask directly for the implicit property, priority, belief, preference,
 intention, relationship, social role, ability, approach, or higher-level
 conclusion.
5. Use an ordinary subject name and, where needed, at most one broad contextual
 anchor. Do not disclose the particular action or statement from which the
 answer is inferred.
6. Require a genuine representational transformation, such as:

 - concrete action -> abstract priority or property;
 - behaviour -> belief, preference, or approach;
 - repeated conduct -> habit or disposition;
 - reciprocal conduct -> relationship state;
 - event participation -> affiliation or interest;
 - concrete outcome -> ability or competence;
 - explicit statement -> intention or attitude;
 - observed decision -> governing value or strategy.

7. The interpretation must be strongly supported by ordinary commonsense. A
 formal logical proof is unnecessary, but a reasonable reader should regard
 the proposed answer as the clear intended interpretation, not merely one of
 several equally plausible possibilities.
8. Scope the interpretation to what the evidence warrants. Prefer questions
 about an approach, priority, role, relationship, or context-specific
 disposition. Do not infer a permanent global personality trait from one
 isolated act.
9. Require the query-memory relationship to be expressible as one short,
 intelligible inferential bridge.
10. Use exactly one complete positive parent per question. The downstream
 preparation stage will use ordinary complete parent-memory negatives; do
 not introduce a positive-versus-negative passage-form difference.
```

```
The intended distinction is illustrated by this example.

Concrete memory evidence:

"Cai Xiuying reviewed her research progress with Li Ming every week, sought his
feedback, and revised her approach accordingly."

Reject an overbroad question:

"Is Cai Xiuying committed to professional growth?"

It makes a potentially permanent personality claim from one episode.

Reject an episode-revealing question:

"What does Cai Xiuying's weekly review of her research with Li Ming suggest
about how she approaches professional growth?"

It copies the concrete behaviour, collaborator, and episode structure into the
query, allowing retrieval without learning the abstraction.

Prefer a question in the following style:

"What professional priority guided Cai Xiuying's approach to improving her
research?"

Supported interpretation:

"Continuous professional improvement and openness to feedback."

This question identifies the broad domain and requested abstract relation but
does not disclose the weekly reviews, Li Ming, or the feedback-seeking behaviour
that supplies the answer.

The query may contain:

- the ordinary subject name;
- one broad domain or context, such as "her research," "the community project,"
 "their friendship," or "his approach to leadership";
- an explicit abstract answer type or relation.

The query must not unnecessarily contain:

- the concrete answer-bearing action;
- the distinctive object involved in that action;
- the name of a secondary participant who fingerprints the episode;
- an exact date, location, project title, event title, product name, or
 numerical specification;
- a rare phrase copied from the memory;
- the supported interpretation or a near-synonymous answer.

If several concrete episode details are required to make the question
unambiguous, reject the candidate and inspect another source case. Do not solve
ambiguity by copying the complete evidence clause into the query.

Reject candidates that:

- merely substitute synonyms or paraphrase the memory at the same level;
- state the abstract answer explicitly in the memory or through a near synonym;
- copy the concrete evidence into the query and append "what does this suggest?";
- infer a broad personality trait from weak or isolated evidence;
- rely on an arbitrary association, stereotype, or obscure specialist knowledge;
- require invented facts or external knowledge;
- admit several substantially different interpretations;
- require information from multiple memories;
- contain multiple independent subquestions;
- leak the answer;
- mention the benchmark, dataset, memory, passage, text, or evidence.

Before accepting a question, perform a semantic ambiguity check against the
other eligible complete memories for the same character. Reject the question
if another memory independently provides an equally plausible answer to the
same abstract information need. This check is only for rejecting ambiguous
questions. Do not choose, record, or optimize against particular negatives, and
do not use embeddings or retrieval scores.

Use 250 distinct source-question IDs and 250 distinct character-qualified gold
parent-memory IDs. Do not treat dialogue and event renderings of the same
underlying episode as independent positives. Do not impose transformation,
character, answer-type, or memory-category quotas. Select solely for Form
validity, semantic support, naturalness, and provenance.

Order eligible source-question and gold-memory pairs deterministically by the
SHA-256 hash of:

 form-mismatch|<source_question_id>|<gold_memory_id>

Review candidates in that fixed order and accept the first 250 satisfying every
rule. If fewer than 250 valid records exist after reviewing the complete
training population, stop and report the shortfall rather than weakening the
criteria.

Write one JSONL record per accepted example containing:
```

```
{
 "question_id": "form-mismatch::0000",
 "source_question_id": "...",
 "character_id": "...",
 "source_question": "...",
 "source_answer": "...",
 "constructed_question": "...",
 "constructed_answer": "...",
 "gold_memory_ids": ["..."],
 "gold_memories": ["exact unchanged complete parent-memory text"],
 "memory_category": "...",
 "underlying_episode_id": "...",
 "concrete_evidence": "...",
 "implicit_target": "...",
 "form_transformation":
"action_property|behaviour_belief|behaviour_preference|action_habit|conduct_relationship|participati
on_affiliation|outcome_ability|statement_intention|decision_priority|other",
 "broad_contextual_anchor": "...",
 "episode_details_omitted": ["..."],
 "inferential_bridge": "...",
 "scope_calibration": "...",
 "same_character_ambiguity_check": "...",
 "validation_notes": "..."
}

Use stable sequential question IDs:

form-mismatch::0000
...
form-mismatch::0249

gold_memory_ids and gold_memories must each contain exactly one item.

concrete_evidence must quote the exact action, statement, event, or behaviour
that supports the answer.

inferential_bridge must explain in one short sentence how the concrete evidence
supports the abstract answer.

scope_calibration must explain why the question does not make a broader claim
than the evidence warrants.

episode_details_omitted must record distinctive evidence details deliberately
excluded from the query. These fields are audit metadata only and must never be
appended to the query, memory, or any encoder input.

Before accepting every record, verify:

- Does the query ask at a meaningfully different level of abstraction?
- Is the abstract answer strongly supported by the unchanged memory?
- Is the abstract answer absent from the memory as an explicit phrase or near
 synonym?
- Does the query omit the concrete behaviour that supplies the inference?
- Does it use only the minimum broad context needed for interpretation?
- Would removing the concrete evidence make the answer unsupported?
- Is the inference appropriately scoped rather than an unjustified global trait?
- Is the question unambiguous within the character's eligible memory store?
- Can the bridge be explained through ordinary commonsense in one sentence?
- Has the complete parent memory remained byte-for-byte unchanged?

Write accepted records incrementally so progress survives interruption. Freeze
the resulting JSONL in a new artifact without overwriting any existing file.

Write a manifest containing the complete prompt, input and output SHA-256
hashes, frozen-split and deterministic-ordering provenance, accepted and
rejected counts, concise rejection reasons, record count, source-question,
positive-parent and underlying-episode reuse, validation results, and
post-freeze distributions of transformation type, character, memory category,
question length, contextual-anchor type, and omitted episode details.

Compute descriptive distributions only after semantic acceptance. Do not alter
the frozen set using retrieval difficulty or downstream information. Do not
proceed to negative mining, training, or evaluation.
```

APPENDIX F.2.4 PROMPT B4 — ENTITY-MISMATCH COMET AUGMENTATION

System prompt The system message is the default Codex agent instruction supplied by the Codex runtime; it is implicit here and is not stored in this repository.

User prompt

```
Construct a frozen set of exactly 250 entity-mismatch questions from
the existing PerLTQA training data.

Inspect only the training questions, answers, labelled gold memory IDs, and
corresponding complete memory texts. Do only question construction. Do not write
a generator or training script, call another language-model API, create or mine
negatives, modify existing code, data, memories, or annotations, train or
evaluate a model, or inspect retrieval rankings, proxy scores, retrieval
```

```
difficulty, downstream data, or downstream results.

For each accepted example:

1. Start from one existing PerLTQA training question and preserve its exact
 answer, character, gold memory IDs, and complete gold memory texts.
2. Rewrite only the question so that a person, event, organization, place, or
 object is referred to differently from how it is expressed in the gold
 memory.
3. Make the query and memory expressions lexically different but clearly
 co-referential from the English question and memory alone.
4. Prefer contextual reference changes such as:
 - "the rally last week" versus "the political activity you participated in";
 - "the physician who treated her" versus the physician's name;
 - "the organization he joined after graduating" versus its explicit name;
 - "the colleague who accompanied her to Beijing" versus the colleague's
 explicit name.
5. Preserve the original information need and requested relation. Do not add a
 second question or require another memory.
6. Reject:
 - translation/transliteration mismatches;
 - unexplained aliases or nicknames;
 - ambiguous pronouns;
 - hidden-metadata relationships;
 - relationships recoverable only because retrieval is already restricted to
 one character's store;
 - outside knowledge;
 - invented facts;
 - descriptions that could identify several referents.
7. Require the original answer to remain fully supported by the exact unchanged
 gold memory.
8. Use exactly 250 distinct source questions. Do not select for retrieval
 difficulty, character, memory category, or any downstream consideration.
 Do not impose per-character, referent-type, or memory-category quotas.

Write one JSONL record per accepted example:

{
 "question_id": "entity-augmented::0000",
 "source_question_id": "...",
 "character_id": "...",
 "source_question": "...",
 "constructed_question": "...",
 "answer": "exact unchanged answer",
 "gold_memory_ids": ["..."],
 "gold_memories": ["exact unchanged complete memory text"],
 "memory_referring_expression": "...",
 "query_referring_expression": "...",
 "referent_type": "person|event|organization|place|object",
 "coreference_justification": "Why the two expressions uniquely identify the same referent"
}

Before accepting each record, verify:

- Can a reader resolve the referent using only the displayed English question
 and memory?
- Do the query and memory clearly concern the same entity or event?
- Is the relationship independent of hidden metadata and store scope?
- Is the original answer still correct?
- Is the information need unchanged?
- Has every memory remained byte-for-byte unchanged?

Continue rejecting invalid candidates until exactly 250 records have been
accepted. Freeze the resulting JSONL before any training or retrieval
evaluation. Write a manifest reporting the empirical distributions of referent
type, character, source-memory category, and source-question reuse. Do not alter
the frozen set in response to those distributions.
```

APPENDIX F.2.5 PROMPT B5 — TEMPORAL-MISMATCH COMET AUGMENTATION

System prompt The system message is the default Codex agent instruction supplied by the Codex runtime; it is implicit here and is not stored in this repository.

User prompt

```
Construct a frozen set of exactly 250 temporal-representation questions from
the existing PerLTQA training data.

Inspect only canonical training questions, answers, labelled gold-memory
references, the corresponding complete rendered parent memories, and repository
code required to resolve and validate those inputs. Do only question
construction, provenance validation, and artifact writing. Do not call another
language-model API, construct or mine negatives, modify source data, train or
evaluate a model, or inspect retrieval rankings, proxy scores, retrieval
difficulty, downstream data, downstream results, previous adapter performance,
or previous generated temporal artifacts.

For each accepted example:
```

```
1. Start from one canonical PerLTQA training question having exactly one
 labelled gold parent. Preserve its exact source-question ID, answer,
 character, canonical namespaced gold-memory ID, and complete rendered gold
 memory text.
2. Rewrite only the question so that its temporal expression differs from the
 temporal representation in the positive memory while identifying the same
 answer-bearing episode, state, or interval.
3. Prefer natural relative, coarse, seasonal, discourse-oriented, or
 event-relative query expressions when the memory represents time
 absolutely. Examples include:
 - "in March 2022," "during the first half of March 2022," or "that spring"
 for a memory dated March 15, 2022;
 - "two weeks after March 1, 2022" for that same date when the arithmetic is
 exact;
 - "after January 2022" or "before April 2022" when the boundary is true;
 - "before their later cinema date" when that event is explicitly stated in
 the query and the relation is unambiguous.
4. Relative-to-absolute matching should be common because it directly targets
 representational mismatch, but do not impose a quota or force an unnatural
 rewrite. Natural absolute-to-absolute and relative-to-relative mismatches are
 also valid.
5. Natural chronology expressions such as before, after, earlier, later, first,
 last, and middle are allowed. Temporal reasoning may help identify the
 positive, but do not manufacture a complex reasoning problem merely to make
 the record difficult.
6. Creation Time is part of the complete rendered embedding key and may supply
 the memory-side temporal evidence. Dialogue timestamps and explicit temporal
 expressions in the natural memory body may also supply it.
7. The temporal cue may be auxiliary to the rest of the question. Do not
 require a same-character wrong-time competitor, and do not require that
 removing the temporal phrase make the question impossible.
8. Preserve the original information need and requested relation. Do not change
 the answer, introduce a second information need, or require another positive
 memory.
9. Keep exactly one complete unchanged positive parent per question. Do not
 split, window, shorten, summarize, concatenate, pad, reconstruct, or rewrite
 any memory.
10. Reject:
 - literal repetition or cosmetic reformatting of the same complete date or
 timestamp on both sides;
 - a temporal phrase that is false, ambiguous, or unsupported;
 - approximate arithmetic presented as exact;
 - direct temporal extraction that merely asks when an explicitly dated
 event happened;
 - invented anchors, invented dates, outside knowledge, or relationships
 recoverable only from hidden metadata;
 - questions whose temporal wording changes the source answer or requested
 relation;
 - noncanonical, ambiguous, or cross-character memory resolution;
 - reconstructed memory text that differs from the canonical rendered text.

Use exactly 250 distinct source questions, 250 distinct character-qualified
positive parents, and 250 distinct underlying episodes. Inspect candidates in a
deterministic source-bound order and stop at the 250th accepted record. Do not
select or balance by retrieval difficulty, character, memory category,
expression family, temporal-pair type, or downstream consideration. Compute
all empirical distributions only after freezing.

Write one JSONL record per accepted example:

{
 "question_id": "temporal-representation::0000",
 "source_question_id": "...",
 "character_id": "...",
 "source_question": "exact unchanged canonical source question",
 "constructed_question": "...",
 "answer": "exact unchanged canonical answer",
 "gold_memory_ids": ["exact canonical namespaced item_id"],
 "gold_memories": ["exact complete rendered canonical memory text"],
 "query_temporal_expression": "exact substring of the constructed question",
 "memory_temporal_expression": "exact substring of the canonical memory or its Creation Time",
 "memory_temporal_evidence_source": "creation_time|dialogue_timestamp|natural_body",
 "temporal_pair_type":
"relative_query_absolute_memory|absolute_query_absolute_memory|relative_query_relative_memory",
 "temporal_compatibility_justification": "Why the two expressions identify compatible times"
}

Before accepting each record, verify:

- Does the source question belong to the frozen training-character split?
- Does the gold reference resolve uniquely inside the correct character's
 canonical memory store?
- Do the recorded gold ID and complete memory text match the canonical item
 exactly?
- Is the query temporal expression an exact constructed-question substring?
- Is the memory temporal expression present in the declared canonical evidence
 source?
- Are the two expressions temporally compatible?
- Is the constructed question natural and unambiguous?
- Is the source answer still fully supported?
- Is the information need unchanged?
- Is this source question, positive parent, and underlying episode unused?

Write accepted records incrementally into a new non-overwriting artifact. Freeze
the JSONL before training or retrieval evaluation. Then write a manifest
```

```
containing the complete construction protocol; canonical input, renderer, and
output SHA-256 hashes; split and deterministic-ordering provenance; accepted
and rejected counts; validation results; and post-freeze distributions of
temporal pair type, expression family, evidence source, character, memory
category, source-question reuse, positive-parent reuse, and underlying-episode
reuse. Do not alter the frozen set in response to those distributions.
```

APPENDIX F.2.6 PROMPT B6 — PLACEBO (GENERIC AUGMENTATION)

The Placebo-1250 dataset is generated through the prompt below. Placebo-250 consists of its first 250 queries, which follow the prompt's fixed SHA-256 ordering of source questions, and Placebo-750 is a random subset of 750 of its queries (seed 42).

```
Construct a frozen set of exactly 1250 questions from the existing PerLTQA training data. The data
is a JSONL file at `...\full_perltqa.jsonl`. Write the newly generated set in the folder named
`comet/rq2/artifacts/placebo` in the source directory.

Do only question construction, provenance validation, and artifact writing. Perform the semantic
judgments yourself in this Codex task. Do not call an external language model or API. Do not write a
general question generator, construct or mine negatives, modify existing code, data, memories, or
annotations, train or evaluate a model, or inspect retrieval rankings, proxy scores, retrieval
difficulty, downstream data, downstream results, previous adapter performance, or existing
generated-question artifacts.

Treat each unique `question_id` as one source question, rather than treating each JSONL row as a
separate candidate. Collapse repeated rows for the same `question_id`. Use `cluster_id` as
`character_id`, `query` as `source_question`, and the complete set of distinct associated
`positive_id` and `positive_turn` values as the gold-memory IDs and exact gold-memory texts. A
source question with multiple gold memories remains one source question with its complete
gold-memory set.

Use the canonical PerLTQA training-case artifact only to copy the unchanged source answer and
validate the complete gold-memory ID set. Read only `case_id`, `answer`, and `gold_item_ids` from
that artifact, joining `case_id` to `question_id`. Do not inspect or use its other fields.

The scientific objective for this frozen set is:

For each question, perform a meaningful rewrite of the question adhering to the following criteria:

1. The rewritten question expresses the same meaning as the original question.

2. The rewritten question contains the same information need as the original question, neither
narrower nor broader, without any additional sub-questions or fewer sub-questions.

3. The rewritten question requires exactly the same ground-truth memory or memories to answer it,
with no other information necessary and no other ground-truth memory required.

4. The rewritten question is not meaningfully more or less lexically similar to the memory; for
instance, it should share roughly the same number of lexicons with the memory as the original
question.

5. The rewritten question should target the same entities, such as people, places, and events, as
the original question unless it is absolutely necessary to introduce new ones, and these entities
should be named identically to the original question.

6. The rewritten question should contain the same time expressions as the original question unless
absolutely necessary in the rewrite to introduce new ones or rewrite them. Time expressions may be
absolute, such as "2025-08-19," or relative, such as "last Sunday."

7. Rewrites should target connective language, phrasing, word order, non-memory-related verbiage, or
grammar.

The intended distinction is illustrated by this example.

Concrete memory evidence: "Cai Xiuying reviewed her research progress with Li Ming every week,
sought his feedback, and revised her approach accordingly."

Original question: "What role did Li Ming play in Cai Xiuying's research?"

Accepted rewritten question: "How did Li Ming help Cai Xiuying's research?"

Rejected rewritten question: "How frequently did Li Ming see Cai Xiuying to help with her research?"

Order eligible source questions deterministically by the SHA-256 hash of:

`placebo|<source_question_id>|<gold_memory_id>`

For a source question with multiple gold memories, use the lexicographically sorted gold-memory IDs
joined by `|` in the `<gold_memory_id>` position.

Eligibility at this stage is determined only by source provenance: the required source fields must
be present, repeated rows must be consistent, and the complete gold-memory set must validate against
the canonical training-case record. Do not apply any semantic, difficulty, or performance-based
eligibility judgment.

Freeze the first 1250 eligible source questions in this fixed order before constructing or reviewing
any rewrites. Construct rewrites for exactly those 1250 questions. Do not skip, replace, or reorder
a frozen source question based on its content or ease of rewriting. If one of the frozen questions
cannot be rewritten while satisfying all seven criteria, stop and report that question rather than
substituting a later candidate.
```

````
Write one JSONL record per completed example containing:
```json
{
 "question_id": "placebo::0000",
 "source_question_id": "...",
 "character_id": "...",
 "source_question": "...",
 "source_answer": "...",
 "constructed_question": "...",
 "constructed_answer": "...",
 "gold_memory_ids": ["..."],
 "gold_memories": ["exact unchanged complete parent-memory text"]
}
```

`constructed_answer` must be exactly identical to `source_answer`.

Use stable sequential question IDs:

`placebo::0000` through `placebo::1249`

Write completed records incrementally so progress survives interruption.

Freeze the resulting JSONL in a new artifact without overwriting any existing file. Write a manifest
containing the complete prompt given to you here, input and output SHA-256 hashes, and frozen-split
and deterministic-ordering provenance.
````

APPENDIX F.2.7 PROMPT B7 — END-TO-END ANSWERING

LoCoMo system prompt

```
You are an intelligent memory assistant tasked with retrieving accurate information from
conversation memories.

# CONTEXT:
You have access to memories from two speakers in a conversation. These memories contain timestamped
information that may be relevant to answering the question.

# INSTRUCTIONS:
1. Carefully analyze all provided memories from both speakers.
2. Pay special attention to the timestamps to determine the answer.
3. If the question asks about a specific event or fact, look for direct evidence in the memories.
4. If the memories contain contradictory information, prioritize the most recent memory.
5. If there is a question about time references (like "last year" or "two months ago"), calculate
the actual date based on the memory timestamp. For example, if a memory from 4 May 2022 mentions
"went to India last year," then the trip occurred in 2021.
6. Always convert relative time references to specific dates, months, or years.
7. Focus only on the content of the memories from both speakers. Do not confuse character names
mentioned in memories with the actual users who created those memories.
8. The answer should be less than 5-6 words.

# APPROACH:
1. Examine all memories that contain information related to the question.
2. Examine their timestamps and content carefully.
3. Look for explicit mentions of dates, times, locations, or events that answer the question.
4. If the answer requires a calculation, perform it before answering.
5. Formulate a precise, concise answer based solely on the evidence in the memories.
6. Double-check that the answer directly addresses the question.
7. Ensure the final answer is specific and avoids vague time references.
```

PersonaMem system prompt

```
Use the persona instructions below and the supplied memories of earlier interactions to respond to
the current user. Choose the single best candidate response from the provided options. Return only
its option label, such as (a), with no explanation.
```

APPENDIX F.2.8 PROMPT B8 — ANSWER JUDGING (LOCOMO)

```
I will give you a question, a correct answer, and a response from a
model. Label the response CORRECT if it contains the correct answer. Otherwise,
label it WRONG. If the response is equivalent to the correct answer or contains
all the intermediate steps needed to obtain the correct answer, label it
CORRECT. If the response contains only a subset of the information required by
the correct answer, label it WRONG.

Return a JSON object with exactly one field, label, whose value is CORRECT or WRONG.

Question: {question}
Question reference date: {question_reference_date}
Gold answer: {gold_answer}
Generated answer: {generated_answer}
```