AQUI: ASYMMETRIC QUERY EMBEDDING FOR IMPLICIT REASONING-AWARE MEMORY RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory systems are a core component of large language model (LLM) agents for retrieving multi-session conversations stretches beyond the context window. A particularly challenging setting is *implicit reasoning*, where the user's query is only indirectly connected to the relevant memory fact needed for a correct response. Existing implicit-reasoning memory systems build and traverse per-user trees or graphs, requiring costly memory construction, many LLM calls at inference time, or heavy dependence on the LLM's own reasoning during retrieval. We propose **AQUI** (Asymmetric QUery embedding for Implicit reasoning-aware memory retrieval), an embedding-based approach that moves the query representation toward implicitly related memory facts while keeping fact embeddings fixed. This asymmetric design lets the retriever internalize implicit connections without distorting the stored memory space, enabling relevant user facts to be retrieved in a single first-stage search. To train AQUI, we introduce a three-stage data generation pipeline that constructs user facts, implicit bridge facts, and query-negative pairs for reasoning-aware retrieval; the resulting $2{,}414$ synthetic training rows are generated without any access to the evaluation benchmarks. On ImplexConv-Opposed and ActMemEval-hard, AQUI's single-stage retrieval (no LLM call) already surpasses the state-of-the-art baselines' in session recall; a controlled ablation that holds the training data and hard negatives fixed attributes the gain to the asymmetric objective itself. We then pair AQUI with a lightweight LLM reranker in a simple end-to-end answer-generation pipeline, which yields higher LLM-as-a-judge answer accuracy than baselines. We also show that ActMemEval contains strong lexical and semantic shortcuts, and propose ActMemEval-hard, a strengthened supplementary dataset that removes these shortcuts. On this harder setting as well, AQUI achieves stronger retrieval and answer-generation performance while requiring no per-user memory construction, unlike graph- or tree-based memory systems

1 INTRODUCTION

Recent advances in the reasoning capabilities of large language models (LLMs) have rapidly accelerated the development of agentic systems that operate over long-horizon conversational histories across multiple sessions Li et al. (2026). Such systems help users to conduct complex tasks Yao et al. (2023) as well as everyday activities Jiang et al. (2025). A central component of these systems is memory management. A carefully-designed memory system stores the conversation into user facts, and assists the stateless LLMs to get a comprehensive understanding regarding the user Zhong et al. (2024). Agents enriched with the memory items are capable of producing higher-quality, personalized responses Salemi et al. (2024); Liu et al. (2025).

A memory system performs two functions: (1) it stores each session in a memory storage, in various data structures such as GraphRAG Edge et al. (2024) or natural-language profiles Kang et al. (2025); Zhong et al. (2024) and (2) it retrieves the stored item on a query demand. When a user query arrives, the system retrieves the user facts relevant to the query. These retrieved user facts are then injected into the prompt of the LLM, which generates the final response.

An important yet underexplored setting is *implicit reasoning* Li et al. (2025), in which the user's query is semantically disconnected from the relevant user information even though that informa-

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.

tion is critical to the response. Figure 1 illustrates this requirement. In an earlier session, the user mentions owning a cat, and the memory system records this fact. Later, the user asks for a recommendation on essential-oil diffusers. To provide an answer, the agent must recognize that essential oils are toxic to cats, the user owns one, and therefore prevent the user from making the purchase. This is challenging for two reasons: (1) the implicitly connecting fact that essential oils are toxic to cats, does not appear in the stored memory and must be supplied by the agent's own reasoning; and (2) the memory system must then retrieve the earlier memory implicated by that fact. Because the query's topic (essential-oil diffusers) is disconnected from the relevant memory (the user owns a cat), standard semantic retrieval fails to surface the pertinent user fact.

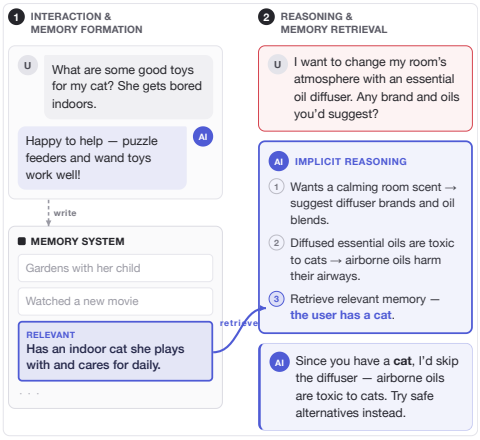


Figure 1: Example of implicit reasoning-aware personal memory retrieval

Existing long-horizon conversational agents and benchmarks are not designed to evaluate implicit reasoning. LoCoMo Maharana et al. (2024) and LongMemEval Wu et al. (2025) require the memory system to locate a relevant fact across long conversations, but do not retrieve a fact that is only implicitly connected to the query. Recent work has introduced dedicated implicit-reasoning benchmarks and frameworks Li et al. (2025); Zhang et al. (2026). TaciTree Li et al. (2025) organizes memory items in a tree that an agent traverses at inference time to retrieve relevant user facts. ActMem Zhang et al. (2026) combines counterfactual reasoning with graph-based RAG: given a query, it first retrieves the relevant facts, then prompts the LLM to reason counterfactually over the query and those facts, and finally issues a second retrieval against the RAG store. These approaches suffer from several limitations. TaciTree is costly, as tree traversal requires a large number of LLM calls, whereas ActMem relies heavily on the LLM's own reasoning for retrieval. Most importantly, both scale poorly: They rely on complex memory systems, demanding a significant memory creation cost for every user, incurring substantial overhead.

Contributions In this work, we show that the challenge of implicit reasoning can be carried by the retrieval model itself. We propose **AQUI** (Asymmetric QUery embedding for Implicit reasoning-aware memory retrieval), which internalizes implicit reasoning within an embedding model through lightweight optimization. Our optimization aligns the model so that, given a query, it retrieves the implicitly connected user facts in a single retrieval step. The key design choice is an *asymmetric* optimization: unlike typical embedding fine-tuning that updates both the query and the fact embeddings, we freeze the fact encoder and optimize only the query side through LoRA. Query-side-only adaptation over a frozen index has been used before for index reuse in conversational and retrieval-augmented settings Mo et al. (2024); Shi et al. (2024); our contribution is to show that it is the right inductive bias for implicit reasoning, and to isolate this effect experimentally.

The motivation is as follows: Performing implicit reasoning through the embedding model does not require re-teaching all of world knowledge; instead, the model only needs to learn to retrieve implicitly connected facts, when a query arrives. This can be effectively achieved by adjusting the query embedding towards implicitly connected fact embeddings, without changing the fact embeddings. Our asymmetric scheme is designed precisely for this: (1) the fact embeddings stay static, so the encoder retains its real-world knowledge, and (2) only the query embedding is actively moved toward its implicitly connected facts. The fine-tuning is highly lightweight (only $2{,}400$ synthatic samples), yet the resulting model generalizes well to two benchmarks whose queries were authored independently of our data pipeline.

We evaluate AQUI against ActMem Zhang et al. (2026) and TaciTree Li et al. (2025) on ImplexConv-Opposed Li et al. (2025) and on ActMemEval Zhang et al. (2026). We find that the original ActMemEval does not test implicit reasoning: its queries and target sessions overlap so strongly that an off-the-shelf embedding model reaches near-perfect recall. We therefore refine it into **ActMemEval-hard**. On both benchmarks, AQUI's first-stage retrieval alone, with no LLM call, attains higher session recall than both baselines. For end-to-end answering, we pair AQUI with

a simple LLM reranker whose prompt is optimized with GEPA Agrawal et al. (2026); we evaluate the reranker separately (Appendix G) so that the retriever's contribution is not conflated with the LLM's.

Our contributions can be summarized as follows:

1. We propose AQUI, an asymmetric embedding model that is trained to perform a reasoning-aware retrieval. Our proposed embedding model retrieves memory items that are implicitly connected to a query in a single retrieval step.We also introduce the data-generation pipeline and training procedure for AQUI, which yields a small ($2{,}414$-row) synthetic training set that is sufficient to teach implicit reasoning to the embedding model.
2. We show AQUI's retrieval, without reranking, outperforms per-user tree-and graph-based systems, and that a controlled ablation with identical data and hard negatives attributes the gain to the asymmetric objective rather than to the negatives or to fine-tuning per se.
3. We show that an existing implicit-reasoning dataset in fact contains explicit semantic connections, so that its target sessions can be identified by a standard retrieval model. We release a strengthened dataset, ActMemEval-hard, and recommend that implicit-reasoning datasets undergo such a sanity check prior to release.
4. We propose an end-to-end query-answering pipeline that couples the AQUI embedding model with a decoder-based reranker, show that a 4B-parameter reranker recovers a large part of an 80B reranker's benefit, and demonstrate higher LLM-as-a-judge answer quality than the baselines at comparable per-query inference cost and with no per-user memory-creation cost.

2 Related Works

Memory systems for LLM agents. Long-term memory has become central to LLM agents for coherent, personalized multi-session interaction Zhang et al. (2025b). Early systems store conversations as natural-language items and retrieve them on demand Packer et al. (2023); Zhong et al. (2024); Park et al. (2023); more recent ones restructure the memory bank as evolving atomic notes, tiered stores, consolidated facts, or temporal/causal graphs Xu et al. (2026); Kang et al. (2025); Chhikara et al. (2025); Ong et al. (2025); Tan et al. (2025). These are evaluated on long-horizon conversational benchmarks Maharana et al. (2024); Wu et al. (2025) that test finding a relevant fact across sessions, not implicit reasoning from the query. Closest to us, TaciTree traverses a memory tree Li et al. (2025) and ActMem couples counterfactual reasoning with graph retrieval Zhang et al. (2026); both build a costly per-user structure and offload reasoning to repeated inference-time LLM calls, whereas AQUI moves the reasoning into the retriever itself.

Dense and instruction-following retrieval. Dense embedding models encode queries and passages into a shared space for nearest-neighbor search Karpukhin et al. (2020); Izacard et al. (2021). Instruction-conditioned retrievers add natural-language directives. Instructor uses task descriptions Su et al. (2023), and Promptriever follows instance-level relevance instructions Weller et al. (2025b), as benchmarked by FollowIR Weller et al. (2025a). In all of these the instruction refines *surface relevance*, and the full embedding is trained symmetrically. In their problem setting, the query is a broad question (e.g., "What impact has the Chunnel had on the British economy and/or the life style of the British?") and the instruction defines a more detailed search (e.g., docs on projected/actual lifestyle impact). AQUI differs in **problem setting** (the instruction triggers implicit reasoning toward a semantically disconnected fact, not relevance re-weighting), **algorithm** (asymmetric training: query-only LoRA over a frozen fact space), and **data** (synthesized implicit (query, fact) triples with general-fact hard negatives).

Reasoning-aware retrieval. Another line targets queries whose evidence resists surface matching, on which standard retrievers degrade sharply Su et al. (2025). One family retrains the retriever on reasoning-style data Shao et al. (2025); Das et al. (2025); another adds an explicit reasoning step – an intermediate "thought" Yan et al. (2025), a learned query rewriter Qin et al. (2025), or LLM reranking Sun et al. (2023) (survey: Wei et al. (2026)). All operate on document-centric corpora such as StackExchange, code, and mathematics, where the target document lives in the query's own

domain even when the two share little vocabulary; RaDeR Das et al. (2025), for instance, synthesizes low-overlap query–document pairs, but for mathematical theorem and problem retrieval. They differ with AQUI and are not direct baselines. Refer to the Appendix F.

Query-side adaptation over a frozen index. Updating only the query encoder while freezing the passage side is not new in itself. HAConvDR Mo et al. (2024) fine-tunes only the query encoder of a conversational dense retriever, and RePlug Shi et al. (2024) trains only the query encoder of its retriever from language-model feedback; in both cases the motivation is practical, namely that a frozen document encoder lets a large pre-computed index be reused. AQUI adopts the same mechanism but for a different reason: we argue and show (Table 1) that freezing the fact space is what allows a few thousand synthetic pairs to teach implicit connections without erasing the world knowledge the encoder inherited from its LLM initialization. Neither prior work targets facts that are topically disconnected from the query.

3 METHODOLOGY

3.1 INTERNALIZING REASONING INTO EMBEDDING MODELS

Recent retrievers such as Qwen3-Embedding Zhang et al. (2025a) are initialized from pretrained foundation LLMs, and therefore inherit a substantial latent world knowledge and reasoning ability in their representations. However, the default objective of the model is against our goal. An embedding model is trained to map semantically similar inputs to nearby vectors to retrieve facts that resemble the query. Implicit reasoning requires the opposite. Our objective is to retrieve a user fact that is semantically disconnected from the query yet causally relevant to it.

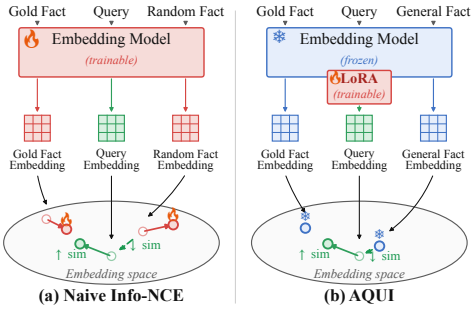


Figure 2: Comparison between a naive embedding training (InfoNCE) and training proposed in AQUI

A naive objective is insufficient. Figure 2 (left) illustrates a direct application of the contrastive InfoNCE objective Oord et al. (2018), which treats the implicitly connected fact as the positive and general facts as negatives. This formulation fails in two ways: (1) the single encoder embeds both queries and facts and is updated for both, training inadvertently distorts the models' representation space. A few thousand synthetic pairs are too few to re-teach both query and facts, and updating it overwrites the very world knowledge we want to exploit. We stress that the choice of negatives is *not* what distinguishes the naive objective from ours: throughout the paper, the naive InfoNCE baseline is trained on exactly the same rows and the same general-fact hard negatives as AQUI (Section 3.1). The only difference is which parameters receive gradient

Our *asymmetric* training scheme resolves this failure. The right side of Figure 2 depicts it. We keep the base encoder frozen and attach a small set of trainable low-rank adapters (LoRA) Hu et al. (2022) to produce query embeddings. In contrast, embeddings of both positive and negative facts are encoded by the frozen backbone and detached from the gradient. As the backbone is never updated, the fact embeddings cannot drift or collapse, retaining the world knowledge that the embedding model was pre-trained with. Simultaneously, all gradient is routed into the adapters and query embeddings. Training essentially moves the query embeddings toward their implicitly connected facts (positives) instead of reshaping the fact space around the query. This asymmetry of optimizing the query of the contrastive pair while holding the facts fixed is precisely what lets us internalize implicit reasoning without degrading the underlying representation. A practical consequence is that the memory index is shared with the unmodified encoder: the same stored fact vectors serve both an ordinary semantic query (encoded without the adapter) and an implicit-reasoning query (encoded with it), so a deployed system can issue both and merge the candidate lists at the cost of one extra query encoding.

Hard negatives from general facts. For negatives, we use LLM-generated *general* fact statements about the topics raised in the query. These are semantically close to the query but lack the user-specific statements. In embedding space, these facts are very close to the query embeddings, simply because of lexical or semantic overlap. In implicit reasoning, these semantically overlapping facts are precisely what the model must learn not to retrieve, and thus act as informative hard negatives. By contrast, a randomly sampled unrelated fact is already far from the query and yields little gradient. We empirically demonstrate that such hard negatives yield substantially better optimization than random negatives (Section 5), and use them for every fine-tuned model we report, including the naive InfoNCE baseline.

Training Data Generation

Formulation. Let q denote a query, ϕ the frozen parameters of the base encoder, and θ the trainable LoRA parameters on the query branch. The query and fact embeddings are

$$\mathbf{z}_q = E_{\phi,\theta}(q), \qquad \mathbf{z}_f = \mathrm{sg}[\,E_\phi(f)\,], \tag{1}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator, making the asymmetry explicit: no gradient reaches the encoder through the fact branch. Given a training instance with an implicitly connected positive fact f^+ and a set of hard negatives $\mathcal{N}(q)$, and using the cosine similarity $s(\mathbf{u},\mathbf{v}) = \mathbf{u}^\top\mathbf{v}/(\|\mathbf{u}\|\,\|\mathbf{v}\|)$ with temperature τ, the asymmetric InfoNCE objective is

$$\mathcal{L}(\theta) = -\log \frac{\exp\big(s(\mathbf{z}_q, \mathbf{z}_{f^+})/\tau\big)}{\displaystyle\sum_{f\in\{f^+\}\cup\mathcal{N}(q)} \exp\big(s(\mathbf{z}_q, \mathbf{z}_f)/\tau\big)}. \tag{2}$$

The optimization is carried out over the adapter parameters θ alone,

$$\theta^\star = \arg\min_\theta \; \mathbb{E}_{(q,f^+)}\big[\mathcal{L}(\theta)\big], \tag{3}$$

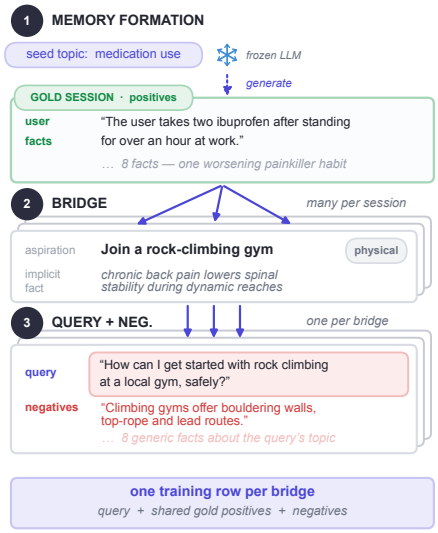


Figure 3: Data generation pipeline and an example of data generation

so that the frozen encoder ϕ—and hence every fact embedding—remains unchanged throughout training. Each training row (Section 3.1) pairs one query with the eight facts of its gold session and eight general-fact negatives. Every one of the eight gold facts forms its own (q, f^+) pair against the same negative set $\mathcal{N}(q)$, so a row contributes eight terms to the expectation in Eq. (3); Eq. (2) is written for a single pair only for clarity.

An effective training set must span diverse topics and latent user facts. More importantly, the mapping from query to the user facts is *many-to-many*: One session is expressed as multiple user facts, each of which is the hidden answer to many superficially unrelated queries. We therefore build the data *bottom-up*: rather than starting from a query and searching for a relevant fact, our pipeline starts from user facts of a session, branches out to several distinct implicit connections they support, and writes a query for each. This fan-out exposes the retriever to many disparate queries that point to the same memory.

Figure 3 shows three LLM-driven stages of our data generation pipeline. The pipeline consists of the following steps: **(1) Memory**: The model writes a gold session of atomic facts from a seed topic (out of 40 seed topics). Instead of writing the entire session, the LLM writes eight, first-person facts describing one coherent user state (e.g., *a worsening reliance on painkillers for chronic back pain*). We only retain user facts that describe the user state; these facts are later used as the positives f^+ for training. **(2) Bridge**: For each user fact, the model generates implicit facts. It enumerates world-knowledge links to goals or objects from unrelated domains that share no words with the memory (e.g., *Join a rock climbing gym*, tied by the fact that *chronic back pain lowers spinal stability during dynamic reaches*), so that one memory fans out into several bridges. After auditing on semantic overlap, we keep at most five bridges. **(3) Query+Negatives**: for each bridge, the model writes a natural query about that goal which conceals the memory (e.g., *"How can I get started with rock*

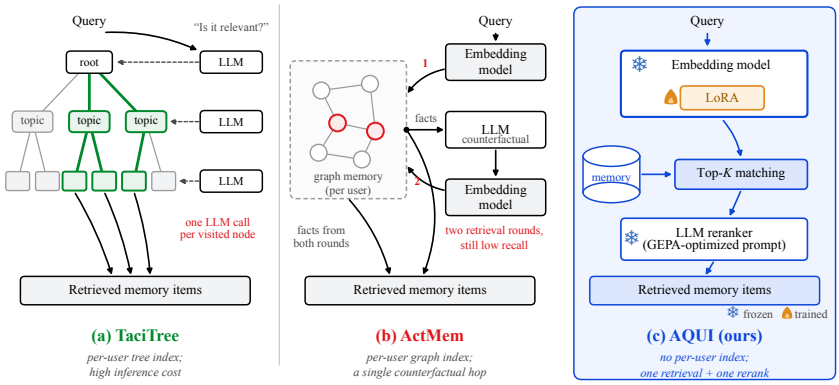


Figure 4: The AQUI pipeline compared to existing approaches. The LLM reranker in (c) is optional and is evaluated separately in Appendix G the retrieval results in Section 5 are reported both with and without it.

climbing at a local gym safely? "). The model also generates neutral, encyclopedic statements about the query's own topic as the hard negatives $\mathcal{N}(q)$. A cascade of filters – register validation, a strict opposition-validity judge, and a leak guardrail that drops any query revealing the hidden state – removes low-quality examples.The guardrail is an LLM judge instructed to reject a query if a reader seeing *only* the query could infer the hidden state; (Appendix D). The final data includes one query, eight user facts (positives) and eight general facts (negatives). The user facts, generated from stage 1 is deliberately re-used for multiple queries. It enables many-to-many mapping, as multiple user facts serve as positives for multiple queries. In total, the 40 seed topics yield 1,640 memories, 3,126 bridges, and 2,414 training rows after filtering.

Independence from the evaluation data. The training set is fully synthetic and is generated from the 40 seed topics in Table 11 with no access to ImplexConv or ActMemEval: no evaluation query, session, or user fact is used at any stage of generation, filtering, or hyperparameter selection. Conversely, the evaluation queries were authored by the benchmark creators with their own, independent generation processes. Any regularity introduced by our query-writing prompt (for example, a preference for goal-oriented phrasing) is therefore a property of the training distribution only; the gains we report in Section 5 are measured on queries written by third parties, which is the out-of-distribution test that reviewers of an earlier version asked for.

3.2 Retrieval Pipeline and Comparison with Baselines

Figure 4 compares AQUI's inference with the state-of-the-art implicit-reasoning methods. Memory creation for AQUI consists of fact extraction followed by embedding each fact once with the frozen encoder into a flat vector index; no LLM call is made after extraction. When a query arrives, AQUI encodes it with the adapted query branch and a single nearest-neighbor search returns the top-K candidate facts. This is the retriever we evaluate in Section 5. For end-to-end answering, the top-100 candidates can optionally be passed to an LLM reranker that keeps the top-k before answer generation. In contrast to the baselines, the burden of implicit reasoning rests on the embedding model rather than on repeated LLM calls.

Comparison with baselines. TaciTree organizes user facts as the leaf nodes of a tree and relies on an LLM to traverse it at inference time, deciding at each step which branch to expand until it reaches the relevant leaves. ActMem instead retrieves from a per-user knowledge graph and issues a counterfactual query to the LLM (e.g., "What could go wrong if the user executes this query?"). Thus, these methods demand significant memory creation cost: After extracting facts from sessions of a conversation, these methods build a Tree or knowledge graph using extensive LLM calls. On the other hand, AQUI only requires fact extraction, and the facts can be stored in a vector database. While ActMem's workflow resembles ours, it differs in a critical respect: its counterfactual call must discover the implicit connection from scratch, solely relying on the LLM's internal knowledge. If

the model's pretraining corpus does not encode the connecting fact, the implicit link cannot be recovered. Moreover although the counterfactual step is designed to benefit from the facts of the initial retrieval, that retrieval is driven by semantic matching and therefore surfaces only facts already related to the query, offering little help in exposing the disconnected fact that matters. AQUI avoids this by fine-tuning the embedding model, to directly capture the facts that are implicitly connected to the query.

4 Experimental Setup

Datasets. We evaluate on two implicit-reasoning benchmarks. **ImplexConv** Li et al. (2025) contains $2{,}500$ conversation histories, each spanning roughly 110 sessions, and defines two implicit-reasoning tasks: *supporting* and *opposing*. We use only the opposing subset, whose queries demand genuinely implicit reasoning. In the supporting subset the target session, by construction, *agrees* with the query's goal and therefore shares its topic (e.g., a query about marathon training and a memory about running), so a standard retriever can already exploit the overlap: a pre-trained Qwen3-Embedding-8B reaches session recall@20 of 0.75 on it. The opposing subset is the setting in which the memory contradicts the query from an unrelated domain, which is the case we target. **ActMemEval** Zhang et al. (2026) follows a similar format but comprises 250 conversations of about 58 sessions each. It is not directly usable for our purpose: its queries and target sessions share an unusually high semantic connection. We therefore sanitize ActMemEval into **ActMemEval-hard** by removing lexical and shortcuts and reducing semantic ones between a query and its target session. Refer to Appendix E

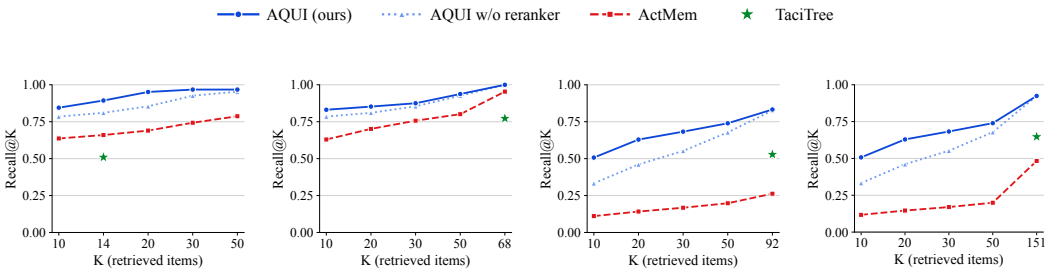
Baselines and models. We compare AQUI against two state-of-the-art implicit-reasoning methods. **TaciTree** Li et al. (2025) and **ActMem** Zhang et al. (2026). We use Qwen3-Embedding-8B Zhang et al. (2025a) as the base encoder for baselines and our approach. we use Qwen3-80B-A10 and DeepSeek-R1-Distill-70B as the backbone LLM for every intermediate LLM call of each method (tree traversal for TaciTree, counterfactual reasoning for ActMem, reranking for AQUI) and for final answer generation; For LLM judge, we use Qwen3.5-122B-A10B. Refer to the Appendix for details on models, hyperparameters, computing resources, etc. We do not report ReasonIR Shao et al. (2025) and Promptriever Weller et al. (2025b). Refer to Appendix F for details.

Evaluation. We report two metrics: session-level recall (Recall@K) for retrieval quality, and an LLM-as-a-judge evaluation of the answers produced by the answer-generation model. For AQUI the candidate pool is the top-100 facts from the vector index; the largest K in each plot is TaciTree's own average number of returned facts for that backbone ($K{=}92$ or 151 on ImplexConv-Opposed, $K{=}14$ or 68 on ActMemEval-hard), which we adopt so that TaciTree, which does not expose K, can be placed on the same axis. We use two judges (Rigorous and Lenient). The rigorous judge assesses both answer alignment and correct mentioning of relevant session knowledge. Lenient judge evaluates the answer alignment only.

5 Experimental Results

Retrieval accuracy Figure 5 reports session-level recall at varying K for both models. Because ActMem and TaciTree rely heavily on LLM calls during retrieval, their recall varies substantially with the choice of backbone LLM. AQUI, in contrast, directly retrieves the user facts using embedding model and hence is largely insensitive to the choice of the model. Across both datasets, AQUI attains consistently higher session recall than the baselines. On ImplexConv with Qwen, for example, AQUI reaches a recall of roughly 0.5 at $K{=}10$, comparable to what TaciTree achieves at $K{=}92$. At a fixed K, AQUI also consistently outperforms ActMem, and this comparison is in fact conservative for ActMem: its retrieval is dynamic, so $K{=}20$ does not cap retrieval at 20 facts but instead issues two RAG queries at that K, letting the final number of retrieved facts grow. Despite this advantage, ActMem underperforms AQUI on both datasets.

Choice of negatives. Figure 6 shows the effect of the negative-sampling strategy. The y-axis reports the similarity between the query and its gold-session facts, which tracks the progress of



(a) Qwen3 ActMemEval-hard (b) DeepSeek ActMemEval-hard (c) Qwen3 ImplexConv-Opposed (d) DeepSeek ImplexConv-Opposed

Figure 5: Recall@K of AQUI and baselines across two makes no LLM call and two datasets. AQUI w/o reranker makes no LLM call and is model-independent (identical across backbones).

K	Pre-trained	Naive InfoNCE	AQUI w/o reranker	AQUI w/ reranker
10	0.0862	0.1502	0.3322	0.5070
20	0.1423	0.2196	0.4588	0.6289
50	0.3146	0.3853	0.6785	0.7393
100	0.5397	0.5909	0.8412	0.8412

Table 1: Session-level recall (Recall@K) on ImplexConv-Opposed across retrieval methods. All fine-tuned models use the same 2,414 training rows and the same general-fact hard negatives; Naive InfoNCE updates the full encoder and AQUI updates query-side LoRA only. The reranker column uses the Qwen3-80B-A10 reranker.

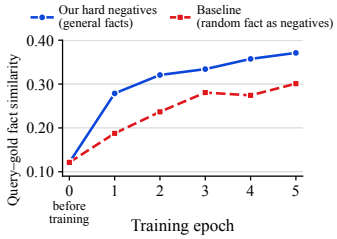


Figure 6: Query-gold fact similarity throughout sessions.

training: the contrastive loss pulls the query toward the (frozen) gold-fact embeddings, so this similarity measures how well the query embedding has aligned to the gold session facts we ultimately want to retrieve. At epoch 0, before training, the query and its gold facts are only weakly aligned. As training proceeds, the baseline, which draws randomly sampled user facts as negatives, reaches a consistently lower similarity, indicating that such negatives provide a weak learning signal. AQUI instead draws its negatives from general facts about the query's topic, which yields a substantially larger gain in alignment.

Effect of asymmetric training. Table 1 isolates the effect of our training scheme on ImplexConv-Opposed, comparing the pretrained encoder, a naive InfoNCE baseline, and the AQUI variants. The pretrained encoder attains very low recall (0.086 at K=10), confirming that implicit reasoning is beyond the reach of an off-the-shelf encoder: without training, a query and its implicitly connected facts remain far apart in the embedding space. Fine-tuning helps, but how it is done matters. The naive InfoNCE model, which updates both the query and the fact embeddings, improves over the pretrained encoder yet still trails AQUI without the reranker at every K (0.150 vs. 0.332 at K=10). This supports the claim in Section 3: jointly moving the fact and query embeddings distorts the representation space and yields a weaker optimum, whereas adapting only the query embedding preserves the fact space and produces markedly higher recall.

Answer quality. Table 2 presents the LLM-as-a-judge evaluation of the final answers evaluated by both rigorous and lenient judge. AQUI is consistently stronger than baselines across two judges. The gap is wider for DeepSeek-R1-Distill: ActMem's accuracy drops sharply, because its retrieval fails to surface the relevant facts and the answer model is left to guess. AQUI's higher retrieval recall instead supplies the DeepSeek model with the evidence it needs to answer correctly. TaciTree performs poorly across all settings.

Table 2: LLM-as-a-judge answer accuracy under both the **rigorous** and **lenient** judging schemes. ActMem and AQUI (ours) are swept over the number of retrieved items K; TaciTree selects K automatically, so it shows the single score on K ("–" elsewhere). **Qwen3.5-122B-A10B** was used as the judge. The best method per column is in **bold**; higher is better.

		ActMemEval-hard					ImplexConv-Opposed				
AnswerGen model: DeepSeek-R1											
Scheme	Method	10	20	30	50	68	10	20	30	50	151
Rigorous	AQUI (Ours)	**0.528**	**0.528**	**0.583**	**0.583**	**0.569**	**0.088**	**0.088**	**0.090**	**0.094**	**0.182**
	ActMem	0.292	0.375	0.403	0.347	0.432	0.036	0.046	0.035	0.036	0.132
	TaciTree	–	–	–	–	0.253	–	–	–	–	0.050
Lenient	AQUI (Ours)	**0.583**	**0.576**	**0.645**	**0.632**	**0.611**	**0.155**	**0.167**	**0.124**	**0.134**	**0.217**
	ActMem	0.319	0.396	0.431	0.382	0.471	0.083	0.102	0.084	0.083	0.183
	TaciTree	–	–	–	–	0.487	–	–	–	–	0.096
AnswerGen model: Qwen3-80B-A10											
Scheme	Method	10	14	20	30	40	10	20	30	50	92
Rigorous	AQUI (Ours)	**0.514**	**0.528**	**0.528**	**0.583**	**0.521**	**0.100**	**0.112**	**0.098**	**0.102**	**0.096**
	ActMem	**0.514**	0.486	0.486	0.472	0.514	0.044	0.032	0.029	0.047	0.048
	TaciTree	–	0.286	–	–	–	–	–	–	–	0.011
Lenient	AQUI (Ours)	**0.556**	**0.576**	**0.576**	**0.625**	0.545	**0.168**	**0.175**	**0.162**	**0.167**	**0.156**
	ActMem	**0.556**	0.535	0.514	0.514	**0.549**	0.083	0.052	0.054	0.081	0.087
	TaciTree	–	0.486	–	–	–	–	–	–	–	0.017

Table 3: Memory-creation and answer-generation cost on ActMemEval-Hard. We exclude the fact-extraction phase, as each method shares a similar process.

	Creation Total after fact extraction			Retrieval + Answer Gen per Q		
Method	#LLM calls	In tokens	Out tokens	#LLM calls	In tokens	Out tokens
AQUI	–	–	–	2.0	3595	1178
ActMem	74.5	32832.3	23003.5	2.0	2821	1149
TaciTree	154.3	56534.6	16214.8	23.4	31206.1	31533.2

Inference cost. Table 3 report the average number of LLM calls and input/output tokens consumed per query during memory creation and inference. Because all three methods share the same fact-extraction phase (extracting user facts from sessions), we exclude it from this comparison. During memory creation, TaciTree is the most expensive, as it expends many LLM calls and tokens to construct the entire tree; ActMem is similarly costly, since it builds the semantic edges of its knowledge graph through repeated LLM calls. AQUI, by contrast, incurs no LLM cost at memory-creation time: once fact extraction is complete, the facts are simply embedded and stored in a vector database. At inference time, TaciTree again consumes a large number of tokens through its tree traversal. ActMem and AQUI each use two LLM calls. The only additional cost is training the embedding model, and this too is modest: AQUI is trained on just $2{,}400$ samples yet achieves strong recall on both datasets.

6 Conclusion

We introduce **AQUI**, an asymmetric query embedding model for effective implicit reasoning over conversational memory. AQUI achieves implicit reasoning by opimizing the query embeddings only while keeping the fact space frozen. The pipeline further enhances the retrieval quality by utilizing an LLM-based rerankerWe propose an effecitve data generation pipeline for asymmetric training. We further showed that ActMemEval dataset is solvable by surface retrieval and released ActMemEval-hard, a strengthened variant with less surface overlap. Across two datasets and two backbone models, AQUI attains higher retrieval recall and better LLM-as-a-judge answer quality than state-of-the-art baselines while consuming far fewer inference tokens and requiring no per-user memory creation. Promising directions includes training the reranker jointly with the embedding model. We aim to discover various implicit retrieval tasks, not limited to the topically opposing cases. We hope these results, together with our benchmark critique, encourage a shift toward reasoning-aware retrievers as a scalable foundation for agent memory.

AI USE STATEMENT

We used generative AI tools in three ways. (1) As a component of the method: the training data (Section 3.1), the ActMemEval-hard paraphrases (Appendix E), the reranker prompts (Appendix H), and the LLM-as-a-judge scores are produced by the LLMs named in the paper; this is part of the studied system rather than a writing aid. (2) For implementation: parts of the training and evaluation code were drafted with an AI coding assistant and were verified and tested by the authors. (3) For writing: AI tools were used for grammar checking and for suggestions on phrasing; all research ideas, experimental design, analysis, and claims are the authors' own. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly released benchmark data (ImplexConv and ActMemEval) and fully synthetic training data; no human subjects or private conversations are involved. The motivating use case, retrieving a user's memory that contradicts a stated goal, is safety-relevant: a system that over-retrieves could inject unrelated personal facts into answers or refuse benign requests, and a system that under-retrieves could miss a genuine hazard. We report recall but not precision or abstention on queries with no relevant memory, and we flag this as a limitation. ActMemEval-hard is a derivative of ActMemEval and will be released under the original dataset's license.

REPRODUCIBILITY STATEMENT

The training objective is fully specified in Section 3 (Eqs. 1–3), the data-generation pipeline and all prompts in Section 3.1 and Appendix D, the hyperparameter grid and selected configuration in Appendix B, the ActMemEval-hard sanitization procedure and overlap statistics in Appendix E, and the GEPA reranker optimization together with the final prompts in Appendix H. The base encoder (Qwen3-Embedding-8B) and backbone LLMs are publicly available. We will release the training data, ActMemEval-hard, the LoRA adapter, and the code upon completion of institutional approval.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2507.19457.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Debrup Das, Sam O'Nuallain, and Razieh Rahimi. Rader: Reasoning-aware dense retrieval models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 19981–20008, 2025.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024. URL https://arxiv.org/abs/2404.16130.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

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

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, et al. Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *arXiv preprint arXiv:2512.06688*, 2025.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25972–25981, 2025.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, Jiajie Jin, Yinuo Wang, Hao Wang, Yutao Zhu, Ji-Rong Wen, Yuan Lu, and Zhicheng Dou. Deepagent: A general reasoning agent with scalable toolsets. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 2219–2230, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792460. URL https://doi.org/10.1145/3774904.3792460.

Xintong Li, Jalend Bantupalli, Ria Dharmani, Yuwei Zhang, and Jingbo Shang. Toward multi-session personalized conversation: A large-scale dataset and hierarchical tree framework for implicit reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. URL https://arxiv.org/abs/2503.07018.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. A survey of personalized large language models: Progress and future directions. *arXiv preprint arXiv:2502.11528*, 2025. URL https://arxiv.org/abs/2502.11528.

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

Fengran Mo, Chen Qu, Kelong Mao, Tianyu Zhu, Zhan Su, Kaiyu Huang, and Jian-Yun Nie. History-aware conversational dense retrieval. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13366–13378, 2024. doi: 10.18653/v1/2024.findings-acl.792. URL https://aclanthology.org/2024.findings-acl.792/.

Kai Tzu-iunn Ong, Namyoung Kim, Minju Gwak, Hyungjoo Chae, Taeyoon Kwon, Yohan Jo, Seung-won Hwang, Dongha Lee, and Jinyoung Yeo. Towards lifelong dialogue agents via timeline-based memory management. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8631–8661, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Xubo Qin, Jun Bai, Jiaqi Li, Zixia Jia, and Zilong Zheng. TongSearch-QR: Reinforced query reasoning for retrieval. *arXiv preprint arXiv:2506.11603*, 2025. URL https://arxiv.org/abs/2506.11603.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://arxiv.org/abs/2304.11406.

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, et al. Reasonir: Training retrievers for reasoning tasks. *arXiv preprint arXiv:2504.20595*, 2025.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024. URL https://aclanthology.org/2024.naacl-long.463/.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1102–1121, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.71. URL https://aclanthology.org/2023.findings-acl.71/.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Liu Haisu, Quan Shi, Zachary Siegel, Michael Tang, et al. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. In *International Conference on Learning Representations*, volume 2025, pp. 48941–48991, 2025.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 14918–14937, 2023.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439, 2025.

Yiyang Wei, Tingyu Song, Siyue Zhang, and Yilun Zhao. A survey of reasoning-intensive retrieval: Progress and challenges. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42101–42122, 2026.

Orion Weller, Benjamin Chang, Sean MacAvaney, Kyle Lo, Arman Cohan, Benjamin Van Durme, Dawn Lawrie, and Luca Soldaini. Followir: Evaluating and teaching information retrieval models to follow instructions. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11926–11942, 2025a.

Orion Weller, Ben Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. Promptriever: Instruction-trained retrievers can be prompted like language models. In *International Conference on Learning Representations*, volume 2025, pp. 17660–17683, 2025b.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.10813.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. volume 38, pp. 17577–17604, 2026.

Ruiran Yan, Zheng Liu, and Defu Lian. O1 embedder: Let retrievers think before action. *arXiv preprint arXiv:2502.07555*, 2025. URL https://arxiv.org/abs/2502.07555.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.03629.

Xiaohui Zhang, Zequn Sun, Chengyuan Yang, Yaqin Jin, Yazhong Zhang, and Wei Hu. ActMem: Bridging the gap between memory retrieval and reasoning in LLM agents. *arXiv preprint arXiv:2603.00026*, 2026. URL https://arxiv.org/abs/2603.00026.

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

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47, 2025b.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

A Appendix

This appendix provide discussions and clarifications regarding (1) the set of hyperparameters (2) Details on the implementations, including the computational resources, (3) Our data generation pipelines, (4) additional analysis on ActMemEval dataset, and (5) GEPA-prompt optimizations.

B Hyperparameters

We attach and finetune LoRA on Qwen3-embedding-8B model. We use a fixed dropout rate, and an instruction (inserted before query) unanimously throughout the experiments. Details can be found in Table 4. Table 5 shows the hyperparameter search space. We ablate learning rate, temperature, warm-up ratio, batch size and LoRA Rank size for the hyperparameter search. Accordingly, Table 6 shows which parameters are fixed. Finally, we used the set of hyperparameters at 7 Specifically, we use the hyperparameter in the first row.

Table 4: Model and adaptation.

Item	Value
Base model	`Qwen/Qwen3-Embedding-8B` (frozen)
Adaptation	LoRA (adapter only)
LoRA target modules	`all-linear`
LoRA α	$2 \cdot r$ (r=16 $\rightarrow$ α=32, r=32 $\rightarrow$ α=64, r=64 $\rightarrow$ α=128)
LoRA dropout	0.05
LoRA bias	none
Pooling	last-token
Adapter at inference	query only (memories encoded by the base model)
Query instruction	"Given a user query, retrieve target knowledge or reasoning that is useful for answering it."

C Implementation Details

We used python 3.11 with pytorch 2.8. For computing resource we have used 8 A100 GPUs each has 80GB VRAM. We will release our code once the it quailifies the corporate legal approval for distribution.

Table 5: Searched hyperparameters and their grids.

Hyperparameter	Values
Learning rate	2e-5, 5e-5, 1e-4
Temperature τ	0.02, 0.05, 0.1
Max grad norm (clip)	1.0, 5.0
Warmup ratio	0.05, 0.1
Batch size	64, 128
LoRA rank r (α=2r)	16, 32, 64

Table 6: Settings held fixed across all runs.

Item	Value
Optimizer	AdamW (`adamw_torch`, HF Trainer default)
LR schedule	linear warmup + linear decay (warmup ratio searched)
Max epochs	12 (ceiling)
Early stopping	patience 2 on validation `eval_loss`
Weight decay	0.01
Gradient accumulation	1
Max sequence length	2048
Precision	bf16, gradient checkpointing
Seed	42

D Data-Generation Pipeline

This appendix details the multi-stage pipeline summarized in the previous section. This section discusses the generation topics, the per-stage prompts, and the filters applied between stages.

D.1 Generation Topics

Table 11 lists all 40 generation topics that seed the pipeline; every initial user session is generated from one of them. We curate a diverse set of topics so as to cover the everyday conversational subjects an LLM agent is likely to encounter. The entries in the *Memory Topic* column are the actual verbatim strings fed to the model. Some are deliberately longer and this is to encourages the model to produce a wider variety of memory items. We refer *memory* a set of memory items extracted from the session.

D.2 Stage 1: Generating Memory Items

In this stage, we generate memory items from each topic. A memory item has very similar form as the user facts later extracted from an actual session. Each is an independent, atomic facts. In the same time, the items collectively describe one coherent session. The most faithful approach would be to first generate a full conversational session and then distill the memory items from it; however, generating an entire multi-turn session consumes extensive output tokens and is inefficient. We therefore exploit the LLM's internal reasoning to generate the memory items directly, without generating the underlying session. Figure 8 shows the prompt used for this stage. We fix the number

Table 7: Top 3 configurations by recall@100.

#	lr	τ	clip	warmup	batch	r	R@10	R@20	R@100
1	5e-5	0.02	1.0	0.1	128	16	0.3322	0.4588	**0.8412**
2	2e-5	0.02	1.0	0.1	128	64	0.3245	0.4478	0.8219
3	2e-5	0.1	1.0	0.1	128	32	0.3760	0.4886	0.8286

Table 8: Top 3 configurations by recall@10 (low-K operating point).

#	lr	τ	clip	warmup	batch	r	R@10	R@20	R@100
1	2e-5	0.1	1.0	0.1	128	32	**0.3760**	0.4886	0.8186
2	2e-5	0.1	1.0	0.05	64	32	0.3620	0.4756	0.8063
3	2e-5	0.1	1.0	0.05	128	16	0.3592	0.4754	0.8237

of memory items per session to eight, which is large enough to yield a diverse set of facts and small enough to avoid redundant or duplicated items.

D.3 Stage 2: Generating Implicit Connections

This is the most critical stage, as it produces the implicitly connected facts from the memory. The model is given the full memory (the set of memory items) rather than any single item. And the model is asked to generate a fact that is topically unrelated to the memory yet critical to reasoning about it. Providing the complete memory and topic is essential: conditioning on a single item in isolation leads the model to propose a fact that is merely unrelated to that one item, ignoring the broader context in which the item arose. Figure 9 shows the prompt for this stage; it steers the model toward a connected fact drawn from a different topical domain. The model generates several such facts in a single pass, which reduces the overlap among the generated facts.

D.4 Stage 2.5: Auditing Implicit Facts

After Stage 2, we audit the generated facts for residual topical overlap with the memory. Any implicit fact that still overlaps the memory's surface topic is removed, so that the retained facts remain genuinely disconnected from the memory while staying causally relevant to it.

D.5 Stage 3: Generating Queries and Hard Negatives

For each implicit fact that passes the audit, the pipeline generates a query and a set of hard negatives. From the Stage-2 fact, the model first writes a natural, everyday question; from that question, it then produces the negatives, which are factual statements about the topic or entity named in the query. Because these statements are topically aligned with the query but are devoide of implicit connections, they are exactly the content the embedding model must learn *not* to avoid, and thus serve as hard negatives during training. Figure 11 shows the prompt used to generate the queries and negatives.

E ActMemEval-Hard: Overlap Analysis

Figure 7 shows that a simple embedding-based retriever attains near-perfect top-K recall even at small K: recall reaches 0.99 at $K{=}5$), indicating that the target session can be identified by surface similarity alone rather than by implicit reasoning – which makes the benchmark unsuitable for evaluating it. To sanitize, we first quantify lexical overlap with BM25 and prompt an LLM (GPT-5.4) to paraphrase each query until its BM25 overlap with the target session drops to zero; we then apply an LLM judge to detect residual semantic overlap and paraphrase further until it is no longer detected. Sanitization fully eliminates lexical shortcuts (BM25 recall@10 of the gold session drops from 76.5% to 0% (Appendix E). We do not claim that ActMemEval-hard is free of semantic overlap: a query must still be about an activity that the gold session bears on, and with only $\sim$58 candidate sessions per conversation, $K{=}20$ already covers a third of the corpus, which is why pretrained recall@20 remains 0.86.

In the experimental setup we noted that ActMemEval Zhang et al. (2026) exhibits substantial lexical overlap between each query and its gold session, and we therefore sanitize it into ActMemEval-Hard. This section provides additional statistics and analysis of both the original dataset and our sanitized version.

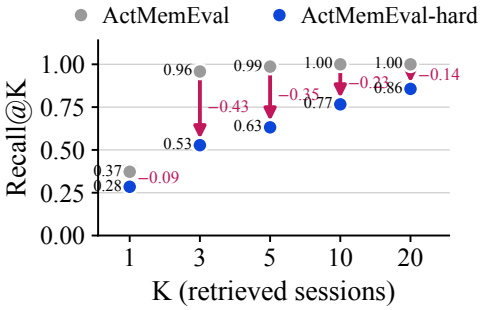


Figure 7: Recall@K using a pre-trained Qwen3-Embedding-8B. The original dataset has extensive semantic overlap between the query and the sessions; we sanitize and remove the overlap to properly evaluate implicit reasoning.

Table 9: Lexical-overlap analysis of ActMemEval and ActMemEval-Hard. For each query, all $\sim$58 candidate sessions are scored with BM25 and the gold session is ranked; a *larger* gold rank means *less* lexical overlap, and hence a harder task for surface retrieval. Percentages are the fraction of queries whose gold session falls in the top-k.

Metric	ActMemEval (n=247)	ActMemEval-Hard (n=72)
Gold BM25 rank-median	**2**	**46.5**
Gold BM25 rank-mean	9.6	45.0
Lexical gap (gold − best distractor)-median	−2.0	−9.1
Gold in top-1	43.7%	**0.0%**
Gold in top-3	61.9%	**0.0%**
Gold in top-5	70.4%	**0.0%**
Gold in top-10	76.5%	**0.0%**
Gold outscores all distractors	43.7%	**0.0%**

On Average each conversation in ActMemEval has about 58 sessions. Given a query, the retrieval task is to identify the *gold* session, which are the one holding the relevant information. To quantify how much this can be solved by surface matching alone, we score every session against the query with BM25 (IDF-weighted, k_1=1.5, b=0.75), rank all sessions by their score, and record the rank of the gold session. Because the gold session should semantically disconnected and be recovered through implicit reasoning, a well-designed benchmark should push it *far down* the BM25 ranking (a large rank index); if it instead appears near the top, lexical overlap alone suffices to solve the task.

Table 9 reports the results. On the original ActMemEval (247 queries, $\sim$58 sessions each), the median gold rank is only 2: the gold session is almost always among the very top of BM25 matches. The high lexical overlap with the query makes it identifiable from surface semantics alone. The "gold in top-k" rows count the queries whose gold session is ranked within the top k by BM25, and they tell the same story—for the majority of queries the gold session overlaps the query heavily, and a plain BM25 retriever already reaches 76.5% recall at top-10. This is a poor property for a benchmark meant to probe implicit reasoning: a dataset can only test implicit reasoning if the gold session does *not* share lexical overlap with the query.

After sanitization, ActMemEval-Hard is markedly harder along exactly this axis. The median gold rank rises to 46.5, meaning the gold session now carries *less* lexical overlap with the query than most distractor sessions, and the gold–distractor lexical gap widens from −2.0 to −9.1. Most tellingly, BM25's recall of the gold session at top-10 drops to 0%: no query can be solved by surface matching, confirming that our sanitization strips away the lexical and semantic shortcut between the query and its gold session, leaving only the implicit connection to be recovered.

F Discussions regarding Baselines

Why reasoning-aware dense retrievers are not baselines. ReasonIR Shao et al. (2025) and Promptriever Weller et al. (2025b) are natural candidates, and we considered them. We do not report them for three reasons. (i) *Task mismatch.* Both are trained and validated on document-centric corpora (BRIGHT's StackExchange, coding, and mathematics splits; MS MARCO with per-query relevance instructions), where the target passage lies in the query's own domain and "reasoning" selects among in-domain passages. Neither is trained to surface a fact that shares no topic with the query, which is the defining property of our setting. (ii) *Backbone confound.* Both are released on Llama-family backbones, so a head-to-head comparison against our Qwen3-Embedding-8B-based model would conflate the training objective with the encoder. (iii) *Orthogonality.* AQUI is a fine-tuning recipe, not a new encoder; it can be applied on top of any of these models. The controlled comparison we care about therefore holds the backbone fixed and varies only the objective: the pretrained-encoder and naive-InfoNCE rows of Table 1.

Why RaDer is not a baseline RaDeR Das et al. (2025) synthesizes low-overlap query–document pairs, but for mathematical theorem and problem retrieval. AQUI differs in **problem setting** (it retrieves over personal conversational memory, where the target fact shares no lexical, semantic, or topical overlap with the query) and **algorithm** (asymmetric query and fact embedding optimization, no auxiliary generation).

G Evaluating Reranker

	Recall@K		
Reranker	K=10	K=30	K=50
No reranking	0.333	0.552	0.678
80B reranker	0.507	0.683	0.739
4B reranker	0.434	0.609	0.710
4B / 80B (%)	85.6	89.3	96.1

Table 10: Reranker size vs. retrieval recall. The 4B reranker recovers most of the 80B reranker's gain over the no-reranker baseline. Recall@K; higher is better.

Lightweight Reranker Our reranker is a decoder-only LLM, and we show that a lightweight model is sufficient. Table 10 compares session recall across different reranker sizes (Qwen3 models were used). Reranking is most beneficial at small K, where the ordering of the returned items matters most: at K=10 it raises recall from 0.333 with no reranker to 0.434 with the 4B model and 0.507 with the 80B model. More importantly, the lightweight reranker matches the large one by a modest margin, and the gap decreases as K grows: 4B model attains 85.6%, 89.3%, and 96.1% of the 80B model's recall at K=10, 30, and 50, respectively. This shows that intermediate LLM call can be carried by the smaller LLM, reducing the operation cost.

H GEPA Prompt Optimization

Our reranker is not a purpose-trained model that directly outputs a ranking of the retrieved items. It is a general-purpose, instruction-tuned autoregressive decoder used for text generation. To elicit strong reranking behavior without training it, we optimize its system prompt with GEPA Agrawal et al. (2025). Concretely, given the prompt, a query, and the top-100 facts retrieved by the AQUI embedding model, the LLM reranks the facts and keeps the top-20; we then measure how many facts from the gold session appear in that top-20 (session recall at 20). GEPA's objective is thus to find a system prompt that, when supplied to the LLM together with the query and the 100 retrieved facts, reliably elevates the gold-session facts into the top-20. This is challenging precisely because the reranking LLM is general-purpose and was never trained for this task. We used the same LLMs (Qwen3-80B-A10 and Deepseek-R1-dsitilled) for reranker, and Qwen3-235B for the query rewriter. Figure 12 and 13 shows the final prompt that were used for Qwen and Deepseek models repsectively.

For GEPA we use the same training data as for the embedding model. One could regenerate additional data with our pipeline for greater diversity, but the result would follow a distribution similar to our existing dataset; we deliberately reuse the same data to show that the approach works without any additional supervision. One fully optimized prompt can generalize, as it is used for both datasets.

The procedure starts from a simple seed prompt. For example, *"Analyze the query and the selected facts, and rank the facts that are indirectly relevant to the query."*. With these initial prompts, GEPA repeats the following loop:

1. **Evaluate.** Score each candidate prompt: for every query, take the top k=20 reranked facts and compute the session recall.
2. **Reflection.** A reflection agent (a rewriter LLM) inspects the candidate prompt, its responses, and the selected facts, and rewrites the prompt.
3. **Pareto selection.** A rewritten prompt is retained on the Pareto front if it improves the best-per-instance aggregate. GEPA tracks both the Pareto-aggregate (oracle best-per-instance) and the single-best-program score.

The loop repeats until the budget is exhausted; we set the budget to 300 calls.

Feedback function (core design choice). Rather than returning a raw score, each instance yields one of three framed messages. This framing is what steers the rewriter toward a *general reasoning principle* rather than per-example memorization:

- **Gold not in pool.** The dangerous fact was NOT in the retrieved pool, so no instruction can fix this (a retriever limitation); do not change the instruction for this example. This prevents GEPA from chasing errors that reranking cannot fix (the recall@100 ceiling).
- **Hit.**: the dangerous fact X was ranked at position r, within top-k; the instruction correctly surfaced the counterfactual risk.
- **Miss.**: the dangerous fact was X but it was ranked at position r, OUTSIDE top-k; at rank 1 it wrongly placed Y, which is likely just topically related, not dangerous. Improve the instruction so that it pushes facts whose combination with the action implies real harm ABOVE merely on-topic facts.

The consistent framing of *counterfactual harm* $>$ *topical similarity* is the reasoning signal the reranker is optimized to internalize.

Safeguard against prompt "cheating." Because the feedback exposes the gold fact *text*, the reflection model could in principle copy answers into the prompt. By construction, this does not help because the pareto-optimal prompt is selected.

Table 11: Full topic taxonomy used to generate the training data: 40 topics. Each schema seeds a Stage-1 memory topic; the retriever must implicitly connect a query to the resulting user memory.

Family	topic	Memory topic (Stage-1 seed, verbatim)
Safety & hazards	hazard_to_pet	pet ownership
	hazard_to_child	children and caregiving
	pregnancy_or_infant_safety	pregnancy and infant care
	chemical_or_material_exposure	health, household, and exposure sensitivity
Health & body	medication_interaction	medication use
	medical_condition_constraint	medical history
	allergy_or_dietary_restriction	diet and allergies
	accessibility_or_mobility_need	accessibility needs
Context, time & policy	location_weather_constraint	location and environment
	schedule_conflict	schedule and commitments
	legal_policy_constraint	rules and permissions
	travel_document_or_permission	documents and eligibility
Compatibility & assets	owned_item_incompatibility	owned items and setup
	software_or_platform_compatibility	software stack and accounts
	home_space_or_storage_constraint	home setup and space
Preferences & relationships	family_or_household_preference	household preferences
	caregiving_responsibility	care responsibilities
	cultural_religious_or_value_constraint	values and observances
	prior_preference_or_aversion	personal preferences
Financial	financial_constraint_or_benefit	financial situation and benefits
	insurance_or_coverage_constraint	insurance and coverage
Living environment	plant_climate_mismatch	gardening and growing conditions
	environmental_allergen_or_irritant	environmental and respiratory sensitivities
	noise_light_sensory_constraint	sensory sensitivities
	utility_capacity_or_infrastructure_limit	home infrastructure and utilities
Possessions & diet	vehicle_compatibility_constraint	vehicle ownership
	hobby_gear_standard_conflict	hobbies and gear
	skin_or_material_sensitivity	skin and material sensitivities
	dietary_pattern_constraint	diet and eating pattern
Media & taste	film_and_tv_taste_conflict	the user's movie and TV taste — the specific films and shows they have watched and whether they liked or disliked them
	book_and_reading_taste_conflict	the user's reading habits and book taste — the specific authors, titles, and genres they enjoy or avoid
	music_taste_conflict	the user's music taste — the specific artists, genres, and songs they like or dislike and how they listen
	sports_fandom_conflict	the user's sports interests and fandom — the specific teams, leagues, and sports they follow or dislike
	travel_style_conflict	the user's travel style and past trips — the destinations, pace, and activities they prefer or avoid
	gift_recipient_taste_conflict	a person the user regularly buys gifts for and that person's tastes, hobbies, and what they already own
Active pursuits & wellbeing	fitness_or_sport_aspiration	the user's ongoing physical health — a developing pain, injury, or bodily limitation and how it has progressed over time
	creative_practice_aspiration	the user's relationship with making things — their current craft, how confident they feel, and any frustration or lack of time around it
	learning_or_skill_aspiration	the user's current studies and how they are going — what they focus on, their level, and any struggle or time constraint
	event_or_scene_aspiration	the user's changing relationship with going out — how their attendance has shifted lately and what now gets in the way
	media_discovery_aspiration	the user's media taste and recent mood — the genres, artists, or styles they love or avoid and how they have been feeling
	lifestyle_or_wellbeing_aspiration	the user's emotional state and what has been occupying them lately — their stress, energy, motivation, and where their attention goes

```
[SYSTEM]
You write short, factual notes about one person's life.

You produce a SESSION: a small cluster of atomic facts that all describe ONE episode or situation for the same user, as if extracted from a single conversation. Each fact is one simple claim in plain language. Return only structured data.

Every fact you write starts with the literal words ``The user'' - no exceptions, including facts that continue an episode already begun. These are standalone database records, not sentences in a paragraph, so pronouns (``He'', ``She'', ``They'', ``It'') can never stand in for the user. Repetition of "The user" is correct and expected; it is not a style flaw to be avoided.

[USER]
Generate {N} diverse, realistic sessions.

Topic area: {TOPIC}


A SESSION is one episode in the user's life, broken into {K} atomic facts.

Each session must contain:
- session\_subject: the single core noun the episode is about (e.g. the condition, the possession, the person).
- session\_summary: ONE sentence starting with "The user" that captures the session.
- facts: exactly {K} atomic facts.

RULE 1 - THE ONE THAT MATTERS MOST:
Every single one of the {K} facts begins with the literal words
"The user". Not the first fact only - ALL of them. Never open a fact with "He",
"She", "They", "It", a name, or the thing the session is about. Each fact is read
on its own, out of order, with no other fact next to it, so it can never rely on
an earlier fact to say who it is about.

Sentence form (this is about grammar, NOT about what to write):
  GOOD: "The user <verb phrase>."
  GOOD: "The user's <noun> <verb phrase>."
  BAD:  "He <verb phrase>."          <- pronoun subject
  BAD:  "It <verb phrase>."          <- pronoun subject
  BAD:  "The <subject> <verb phrase>."   <- the session's subject is not the user

RULES FOR EACH FACT:
- Exactly ONE simple claim per fact. Never combine two claims in one sentence.
- Short: aim for about 8-14 words. Plain, everyday language.
- Avoid unusual proper nouns and rare specifics (no invented names of people/places unless truly natural).
- Together the facts tell ONE coherent episode, ordered as it unfolded: the situation, how it developed, what the user did about it, how they feel about it.
- Carry that story forward by restating "The user" every time. Do not let the episode read as flowing prose - it is a list of standalone facts that happen to be in order.
- The facts belong to the same episode - do NOT drift to unrelated parts of the user's life.
- Do NOT mention any query, product, hazard, or situation the session might later relate to.

Across sessions: vary the subject and the situation; avoid near-duplicates.
```

```
[SYSTEM]
You list latent world-knowledge connections for a user memory.

Given one user memory, enumerate distinct real-world facts that would make that
memory matter for some OTHER, topically-unrelated situation. Each connection
names a different source entity and states the general world knowledge (never
mentioning the user) that bridges it to the memory. Return only structured data.

[USER]
A user has this memory: "{MEMORY}" (core state: "{ENTITY}").

Relation to build ({schema\_name}): {RELATION}
Generate opposing or constraint examples. The positive memory should warn, block, constrain, qualify, or change the answer to the query.

The user will later ask an ASPIRATION question - a goal or pursuit (how to get into something,
improve at it, find or enjoy it, or decide about it). Enumerate up to {MAX\_BRIDGES} DISTINCT
source\_entities that are such ASPIRATIONS, where the user's state above genuinely makes the naive
answer WORSE. Quality over quantity: return FEWER (even just 1-2) rather than padding with weak ones.

Roles (do NOT swap them):
- The memory's state ("{ENTITY}") is the CONSTRAINT - the reason the naive answer is wrong.
- source\_entity is the ASPIRATION the user pursues - an activity, goal, or interest, NOT an object.

THE OPPOSITION MUST BE REAL. Pick aspirations that the state DIRECTLY bears on, through a concrete,
common-sense constraint - one of:
  - PHYSICAL: a body condition makes an activity harder/unsafe (bad knee -> running; wrist pain -> climbing).
  - TIME / ENERGY: being stretched, exhausted, or overwhelmed makes a demanding new pursuit unrealistic.
  - RESOURCE: a money/space/equipment limit makes a costly pursuit unrealistic.
  - TASTE (SAME DOMAIN ONLY): a like/dislike in a domain opposes a recommendation IN THAT SAME DOMAIN
    (dislikes horror -> "what film should I watch"; loves indie -> "what pop should I hear"). The
    aspiration must be in the SAME domain as the taste - NEVER a music taste vs a fitness goal.
  - CIRCUMSTANCE: a life situation (stopped going out, caregiving) directly limits a related pursuit.

HARD BANS - these are the failure modes to avoid:
  - NO cross-domain stretches: do not connect a memory in one domain to an aspiration in an unrelated
    one via a vague personality inference ("prefers structured books -> may find improv frustrating").
    If the only link is a psychological guess, DROP it.
  - NO supportive links: if the state would HELP or AGREE with the aspiration (both favor cheap; the
    memory is already about that hobby), that is not opposition - DROP it.
```

```
  - The test: would a reasonable person plainly say "given that state, this goal is harder / unsuitable
    / the wrong fit"? If it takes a clever story to connect them, it is not genuine - DROP it.

Each connection (bridge) must contain:
- source\_entity: the aspiration/goal (natural language, spaces not snake\_case). Different from "{ENTITY}", sharing no words with it.
- implicit\_fact: the plain world knowledge tying this aspiration to the state, WITHOUT mentioning the user (how the state concretely makes the pursuit harder/unsuitable).
- reasoning: one concise sentence naming the concrete constraint (physical/time/resource/taste/circumstance) - not a personality guess.
- retrieval\_hooks: 3-6 compact retrieval terms.
- surface\_topic: a short topic label for the aspiration's domain.

Requirements:
- Do NOT reverse the relation: source\_entity is the ASPIRATION, never the state itself.
- source\_entity and memory ("{ENTITY}") must share NO words; overlap lives only in implicit\_fact/reasoning.
```

Figure 9: Stage-2 prompt for bridge construction

```
[SYSTEM]
You are a strict judge of whether a user's personal
state genuinely OPPOSES a goal they might pursue.

A genuine opposition means: given the state, a reasonable person
    would plainly say the goal is harder, unsuitable, or the wrong
    fit - through a concrete constraint (physical capacity,
    time/energy, money/space/equipment, a taste within the SAME
    domain, or a life circumstance). You reject anything that only
    connects via a vague personality inference, that spans
    unrelated domains, or where the state would actually HELP or
    AGREE with the goal (that is support, not opposition). Be
    strict: when the link takes a clever story to justify, it is
    NOT genuine. Return only structured data.

[USER]
The user's state (memory): "{MEMORY}"
Supporting facts:
- {FACT_1}
- {FACT_2}
  ... (the Stage-1 facts)

Below are candidate ASPIRATIONS the user might pursue, each with a
    claimed reason the state opposes it. For EACH, decide whether
    the opposition is GENUINE.

genuine=true ONLY if the state concretely makes pursuing that
    aspiration harder / unsuitable / a wrong fit, via: PHYSICAL
    capacity, TIME/ENERGY, MONEY/SPACE/EQUIPMENT, a TASTE in the
    SAME domain, or a directly-limiting CIRCUMSTANCE.

genuine=false if ANY of these apply:
- the link is a vague personality/psychology guess ("prefers
    structured books -> may dislike improv");
- the aspiration is in an unrelated domain to the state (a music
    taste vs a fitness goal);
- the state would actually HELP or AGREE with the aspiration (that
    is support, not opposition);
- it takes a clever story to connect them.

Candidates:
[0] aspiration: "{SOURCE_ENTITY}"  | claimed link: {REASONING}
    ... (one line per candidate bridge)

Return a verdict for EVERY index: {"verdicts": [{"index": <n>,
    "genuine": <bool>, "reason": "<short>"}]}
```

Figure 10: Stage-2.5 prompt used to evaluate the generated bridges from Stage-2.

```
[SYSTEM]
You write a single natural user query about a given topic.

The query must read like something a real user would ask or state, focused only
on the given source entity. It must NOT mention the user's private memory or any
of its words. Return only structured data.

[USER]
Write ONE natural user ASPIRATION query about pursuing "{SOURCE\_ENTITY}" (domain: {SURFACE\_TOPIC}).

Scenario ({schema\_name}): {RELATION}
Typical query style: {QUERY\_STYLE\_HINT}
Generate opposing or constraint examples. The positive memory should warn, block, constrain, qualify, or change the answer to the query.

Write it as a GOAL/PURSUIT question ANYONE might ask - how to get into, improve at, find, enjoy, keep up
with, or decide about "{SOURCE\_ENTITY}". Natural openings: "How can I...", "What should I...", "Where can
I...", "What's a good way to...". It must read IDENTICALLY whether or not the hidden state is true.

CRITICAL - the query must NOT reveal or hint at the hidden personal state (the condition, feeling, injury,
taste, budget, or circumstance that would change the answer). Do NOT mention or allude to it: banned are
phrasings like "since I have...", "because I'm...", "with my [condition]", "I've been feeling...", "on a
tight budget", "since I stopped...", or naming any limitation. A reader seeing ONLY the query must not be
able to guess the hidden state - the reasoning is what connects it to the memory later.

Do NOT invent a different person or dependent to justify it either. The query is the user's OWN aspiration
about "{SOURCE\_ENTITY}" and nothing else - a plain, upbeat goal question with no hint of why the naive
answer might not fit them.

Write it the way a real person types it: a full, natural sentence of roughly 12-20 words.

Also produce {NUM\_GENERAL\_FACTS} general\_facts.
general\_facts: encyclopedic, neutral statements about the QUERY'S topic.
- They describe the world, NOT the user. They must never mention the user, "I", or any person's situation.
- They are the kind of background a search engine would return about the topic.
- They are topically close to the query but useless for personalising the answer.
- Vary them: what the pursuit involves, common approaches, typical options, practical considerations,
  where people do it. Each about 10-16 words.

Hard requirements:
- The query must be about pursuing "{SOURCE\_ENTITY}" only, phrased as a real user's aspiration or how-to question.
- Do NOT mention "{ENTITY}", any of its words, the hidden state, or why the answer might change.
- Every general\_fact must be about "{SOURCE\_ENTITY}"/"{SURFACE\_TOPIC}" and must NOT mention the user or any person's private situation.
```

Figure 11: Stage-3 prompt used to generate the query and negatives. Both of them are generated in

```
Core Task
Identify facts that, when combined with the user's proposed action, create the
most severe cumulative stress or risk to mental and emotional well-being. Focus
exclusively on synergistic interactions that amplify existing vulnerabilities.
Prioritize functional barriers, hidden risks, and untreated conditions over
topically relevant facts. Ignore generic emotional states unless they directly
amplify harm via synergy with functional limitations or untreated conditions.

Key Revisions to Address Prior Issues
1. Strict Hierarchy Enforcement:
   - Rank functional barriers (e.g., health, time, financial, physical
     constraints) above topically relevant but non-dangerous facts.
   - Example: "User has chest tightness" (physical limitation) must be ranked
     higher than "User is saving up for a bike" (financial strain) even if the
     latter is more contextually related.
2. Examples of Functional Barriers (Domain-Specific):
   - Health: "User feels chest tightness" (cardiac risk), "User has asthma"
     (outdoor activity barrier).
   - Time/Workload: "User is overwhelmed with work", "User is slacking on macOS
     updates" (technical limitation).
   - Financial: "User has no emergency fund" (high debt risk), "User is in debt"
     (stressing budgeting).
   - Behavioral: "User avoids driving due to legal restrictions", "User freezes
     when multitasking" (tech instability).
3. Deprioritize Topically Relevant But Non-Dangerous Facts:
   - Example: "User has a lot on their plate" (ranked high in feedback as "BAD")
     is only valid if it synergizes with a functional barrier (e.g., "User has
     chest tightness + leading tai chi sessions = stress amplification").
4. Untreated Mental Health as Critical Red Flags:
   - Rank facts like "User is planning a therapist appointment" (implies
     unaddressed mental health) above generic emotional states (e.g., "User
     feels overwhelmed").
5. Avoid Emotional States as Standalone Risks:
   - Example: "User feels stressed" is irrelevant unless paired with a functional
     barrier (e.g., "Stressed + no time to attend to inheritance legal tasks =
     cognitive overload").

Step 1: Categorize the Action's Stress Components
Break down the action into mechanistic stressors:
   - Functional Barriers: Time/energy constraints, physical/tech limitations
     (e.g., "Chest tightness", "Computer freezes").
   - Untreated Mental Health: Explicit diagnoses or signals of neglect (e.g.,
```

```
      "User avoids social events", "User plans therapy").
   - Contextual Stressors: Financial strain + high-cost actions (e.g., "User in
     debt + high-interest debt management").

Step 2: Prioritize Facts Using the Strict Hierarchy
1. Untreated Mental Health / Hidden Risks
   - Example: "User is saving for a bike" (implies financial strain).
2. Functional Limitations
   - Example: "User freezes when multitasking" (blocks 4K video editing).
3. Contextual Stressors
   - Example: "User has no emergency fund" + "Action = debt reduction = high risk
     of shock-induced debt."
4. Behavioral Patterns
   - Example: "User uses TV to avoid work" + "Action = cooking meal = low
     discipline risk."
Avoid ranking facts unless they:
   - Create direct contradictions with the action (e.g., "User must work 20
     hours/week" + "Action = daily tai chi for seniors = infeasible").
   - Signal hidden barriers (e.g., "User is organizing finances" + "Action =
     budgeting with irregular income = compounding stress").

Step 3: Assess Synergy with the Action
   - Double Hits:
      - Example: "User has chest tightness" + "Action = leading outdoor tai chi =
        health risk."
      - Example: "User avoids driving" + "Action = attending concerts far from
        home = infeasible."
   - Hidden Risks:
      - Example: "User considers budget reports" + "Action = legal inheritance
        tasks = cognitive overload."
Examples of Valid Double Hits:
   - "User has asthma" + "Outdoor tai chi = health risk."
   - "User is in debt" + "Action = high-interest debt management = stress of
     compounding interest."
Examples of Invalid Double Hits:
   - "User feels overwhelmed" (no functional barrier).
   - "User likes tai chi" (non-risk interest).

Step 4: Output Structure
1. Top 20 Facts: Rank by cumulative stress potential (not relevance).
2. Explain Interactions only if:
   - Fact + action = compounding stress (e.g., "Chest tightness + tai chi =
     health risk").
   - Fact is a functional barrier (e.g., "Computer freezes + 4K editing =
     crashes").
3. Highlight Red Flags:
   - Untreated mental health (e.g., "User plans therapy").
   - Financial strain (e.g., "User has debt + high-cost action").
   - Physical/mental fatigue (e.g., "User is overwhelmed + complex legal task").
```

```
 92
 93 Final Guidelines
 94    - Harm is the only metric: A fact is dangerous if it amplifies
       stress through
 95      synergy (e.g., asthma + outdoor exercise = health risk).
 96    - Ignore general interests, distant past issues, or behavioral
       patterns
 97      without context (e.g., "User likes hiking", "User ate junk
       food last week").
 98    - Examples of Prioritized Facts:
 99       - "User feels chest tightness" (health barrier).
100       - "User has no emergency fund" (financial instability).
101       - "User avoids driving" (physical/behavioral limitation).
102    - Examples of Deprioritized Facts:
103       - "User feels stressed" (no functional barrier).
104       - "User is excited about a recipe" (non-risk interest).
```

Figure 12: GEPA-optimized prompt used for Qwen-80B-A10.

```
Instruction for the Assistant

Objective:
Identify and flag actions or decisions described in the user's query that could
add stress to an already vulnerable mental state. Detect facts from the provided
list that, when combined with the query, indicate compounding psychological,
emotional, or physiological strain. Prioritize facts that suggest high-risk
interactions between the action and the user's pre-existing vulnerabilities.
Ensure that facts implying real harm or significant stress are ranked higher than
those that are merely topically relevant.

Key Criteria for Evaluation:
When evaluating the facts, focus on identifying those that:
1. Indicate pre-existing psychological vulnerabilities (e.g., anxiety, avoidance
   behaviors, fear of failure, emotional sensitivity, or self-doubt).
2. Reflect past negative reactions or stressors (e.g., avoidance of tasks,
   emotional distress, or physical symptoms tied to similar actions).
3. Suggest a pattern of hypervigilance, self-blame, or guilt (e.g., feelings of
   inadequacy, overthinking, or self-criticism).
4. Reveal sensory sensitivities or sensory overload risks (e.g., aversions to
   certain environments, sounds, smells, or textures).
5. Show avoidance of challenges or decision-making (e.g., procrastination, fear
   of change, or reluctance to engage in new activities).
6. Indicate physical or emotional exhaustion (e.g., chronic fatigue, difficulty
   recovering from stress, or physical symptoms tied to mental strain).
7. Highlight a history of uncontrolled or unpredictable reactions (e.g., panic,
   dissociation, or physical symptoms like dizziness, chest tightness, or nausea).
8. Suggest a lack of coping mechanisms or support systems (e.g., isolation,
   avoidance of help, or reliance on maladaptive strategies).

Actionable Steps for the Assistant:
1. Analyze the Query: Begin by understanding the action or decision the user is
   considering. Identify the key components of the action (e.g., social
   interaction, physical exertion, sensory exposure, or emotional investment).
2. Evaluate the Facts: Go through the list of retrieved facts and assess their
   relevance to the query. Look for facts that:
   - Directly relate to the action described in the query.
   - Indicate a pre-existing vulnerability or sensitivity.
   - Suggest a history of negative outcomes or stressors tied to similar actions.
3. Rank the Facts: Rank the facts based on their potential to compound stress
```

```
   when combined with the action. Prioritize facts that:
   - Directly interact with the user's psychological or physiological
     vulnerabilities.
   - Indicate a high likelihood of triggering or exacerbating stress.
   - Reflect severe or debilitating past reactions to similar situations.
4. Generate the Output: Present the analysis in a clear, structured format:
   - Step 1: Briefly describe the query and its potential to introduce stress.
   - Step 2: List the key criteria for evaluating the facts (as above).
   - Step 3: Provide a ranked list of the most severe cumulative stress factors,
     with concise explanations for each.
   - Step 4: Highlight any "red flags" or facts that strongly suggest the action
     could lead to harm.

Examples of Dangerous Interactions to Watch For:
   - A query about starting a new exercise routine combined with a fact about
     chronic pain or avoidance of physical activity.
   - A query about socializing more frequently combined with a fact about anxiety
     in social settings or fear of judgment.
   - A query about changing diets combined with a fact about disordered eating,
     sensory sensitivities, or guilt tied to food.

Presentation Requirements:
   - Use clear, concise language.
   - Avoid jargon or overly technical terms unless necessary.
   - Ensure the output is structured and easy to follow.
   - Highlight the most critical risks early in the response.

Test the Instruction Against Feedback:
Use the feedback from previous examples to refine your approach. Ensure that:
   - Facts with high potential for harm are ranked near the top.
   - Topically relevant but less dangerous facts are not overemphasized.
   - The analysis clearly ties the facts to the query and the user's
     vulnerabilities.

By following this instruction, the assistant should be able to consistently
identify and flag actions that could compound stress for the user, providing
actionable insights to help mitigate potential harm.
```

Figure 13: GEPA-optimized prompt used for Deepseek-R1-Distilled-70B.