USER QUERIES SAY MORE WITH LESS: QUERY-ANCHORED, LLM-AWARE RESPONSE SELECTION FOR LONG-TERM CONVERSATIONAL MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

In long-term conversational memory, historical user queries and LLM responses exhibit a pronounced asymmetry: user queries are concise and information-dense, whereas LLM responses are substantially longer but often contribute limited new information. We find that historical user queries serve as the primary memory basis, while LLM responses are largely compressible but conditionally useful, with their utility depending on relevance to the current query and the needs of the downstream LLM. Based on these findings, we propose **QARS** (**Q**uery-**A**nchored **R**esponse **S**election), a context selection framework that retains historical user queries as memory anchors and selectively incorporates historical LLM responses. It first filters LLM response candidates by their semantic relevance to the current query, thereby narrowing the candidate set; it then performs downstream-LLM-aware selection over the remaining candidates using trained discriminators conditioned on a behavior-based representation of the downstream LLM. Experiments across three long-term conversational memory datasets and six downstream LLMs show that QARS substantially reduces input tokens while maintaining or improving answer accuracy relative to full retrieved contexts, and outperforms existing compression baselines in accuracy. It further constructs LLM-specific contexts and adapts to unseen downstream LLMs using a calibration set without retraining.

1 INTRODUCTION

In sustained conversations with large language models (LLMs) (Singh et al., 2025; Comanici et al., 2025), user queries and LLM responses alternate over time yet form asymmetric information units: user queries are typically concise and information-dense, whereas LLM responses are substantially longer but often contribute limited new information to downstream LLMs. As shown in Figure 1(a), LLM responses account for 80.2% of tokens in full conversation histories on average. This asymmetry also reflects their different informational roles. User queries serve as the primary entry point for external information, introducing user-specific facts, needs, and intents (Li et al., 2024; Zhao et al., 2025; Li et al., 2026). They also drive how the interaction unfolds: what an LLM generates depends largely on how the user queries it, making LLM responses elaborations of the queries based on the LLM's own knowledge rather than independent sources of external information (Petroni et al., 2019; Jiang et al., 2020).

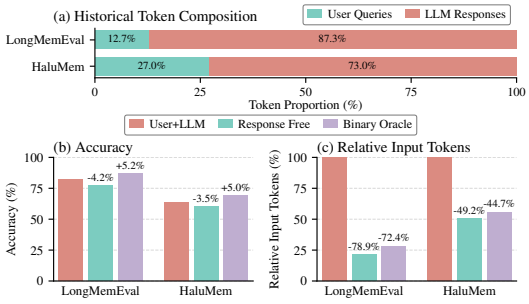


Figure 1: LLM responses account for 87.3% and 73.0% of full-history tokens in LongMemEval and HaluMem, respectively. Under turn-level RAG with Qwen3.5-27B as the downstream LLM, removing historical LLM responses substantially reduces input tokens with only a modest loss in accuracy. Binary Oracle reveals accuracy gains by selecting, for each query, whether to keep or remove all retrieved historical responses.

The query-response information asymmetry above motivates a reconsideration of long-term conversational context management. Existing approaches typically process user queries and LLM re-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

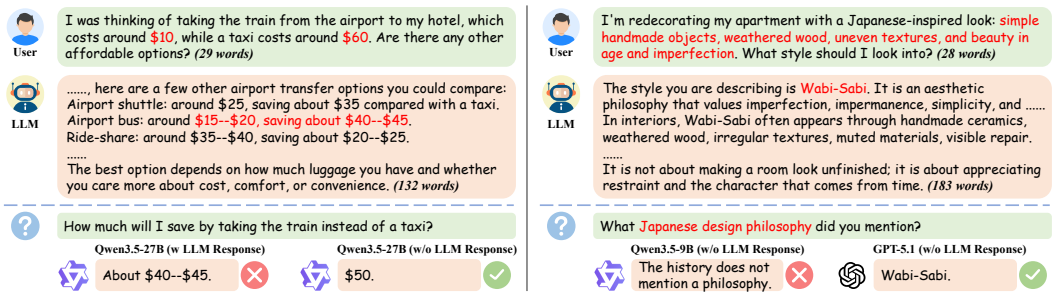


Figure 2: Illustrative cases of historical LLM response utility. **Left**: When a historical user query provides sufficient evidence, the LLM response can be redundant and introduce interference. **Right**: When explicit evidence is absent from the historical user query but appears in the LLM response, GPT-5.1 can recover the information from the user query alone, whereas Qwen3.5-9B cannot.

sponses together, including memory retrieval (Zhong et al., 2024; Pan et al., 2025; Wu et al., 2025; Xu et al., 2025) and context compression (Jiang et al., 2024; Pan et al., 2024; Xu et al., 2024; Hwang et al., 2025). Such joint treatment retains substantial context from lengthy LLM responses that may provide limited additional information, creating an opportunity to center memory on historical user queries while substantially reducing the context contributed by LLM responses. This raises a fundamental question: *how necessary are historical LLM responses in long-term conversational memory?*

As shown in Figure 1, we compare two context configurations: **User+LLM**, which retains retrieved user queries and their LLM responses, and **Response Free**, which removes all LLM responses. Removing LLM responses leads to only a modest accuracy drop of 3.9% on average, while substantially reducing input tokens by 64.1%. We further construct a **Binary Oracle** that retrospectively selects the better configuration for each current query. Compared with User+LLM, it improves accuracy by 5.1% while reducing input tokens by 58.6%. Together, these results suggest that historical user queries serve as the primary memory basis, whereas LLM responses are largely compressible but conditionally useful. This motivates selectively retaining historical LLM responses rather than uniformly keeping or removing them.

As illustrated in Figure 2, the utility of historical LLM responses depends on two factors. **First, some historical LLM responses contain information useful for answering the current query, while others do not.** When the historical user queries already provide sufficient evidence, historical LLM responses with limited relevance may be redundant or even introduce interference. **Second, whether an LLM response is necessary depends on the downstream LLM.** For some downstream LLMs, even historical responses containing explicit evidence may be dispensable if they can recover the information from the historical user queries using their own capabilities. For others, historical LLM responses that add little explicit evidence may still be useful by helping organize, disambiguate, or reason over the user query information. Consequently, highly relevant historical LLM responses may be unnecessary for some downstream LLMs, while less relevant ones may remain useful for others. Thus, the value of historical LLM responses depends not only on their relevance to the current query, but also on whether the downstream LLM needs them.

Motivated by these observations, we propose **QARS**, a user-**Q**uery-**A**nchored, downstream-LLM-aware framework for historical LLM **R**esponse **S**election. The framework treats historical user queries as stable memory anchors and their associated LLM responses as optional candidates, selecting LLM responses according to both their semantic relevance to the current query and their necessity for the downstream LLM. In **Stage 1: Semantic Relevance Filtering**, we assess LLM response candidates based on their relevance to the current query, using their incremental semantic contribution to filter out candidates that clearly reduce relevance while retaining the remaining ones for further evaluation. In **Stage 2: Downstream-LLM-Aware Response Selection**, we determine whether the remaining candidates are useful to the downstream LLM. In particular, a retention discriminator examines whether candidates that increase semantic relevance remain necessary, while an addition discriminator identifies semantically uncertain candidates that provide complementary information the downstream LLM cannot reliably derive or utilize from the user query evidence alone. Both are conditioned on an LLM representation together with the current query, historical user query, and LLM response candidate. We obtain the LLM representation from a lightweight diagnostic calibration set that characterizes how different downstream LLMs use historical informa-

tion under the same query-context configurations. For an unseen downstream LLM, a single pass over this calibration set produces its representation without retraining the discriminators.

Our main contributions are summarized as follows: (i) **We systematically investigate the role of historical LLM responses in long-term conversational memory and reveal a pronounced query-response asymmetry.** Historical user queries and LLM responses differ substantially in both length and informational role, while historical LLM responses are largely compressible but conditionally useful: their value depends on both their relevance to the current query and the needs of the downstream LLM. (ii) **We propose QARS, a user-query-anchored, LLM-conditioned framework for historical LLM response selection.** It first filters LLM response candidates according to their relevance to the current query and then selects the remaining candidates according to the needs of the downstream LLM using behavior-based LLM representations. (iii) **Extensive experiments demonstrate consistent effectiveness across three long-term conversational memory datasets.** Across both turn- and session-level retrieval, QARS improves over RAG while using fewer input tokens and outperforms compression baselines in accuracy. Across six downstream LLMs, QARS shows consistent gains over baselines while selecting historical responses according to the downstream LLM. Ablations validate the two-stage design and LLM representation, and QARS further adapts to unseen downstream LLMs using a calibration set without retraining.

2 Related Work

Conversational Memory Management. Existing approaches to long-term conversational memory primarily adopt long-context modeling or external memory management. Long-context methods feed long conversation histories directly into the model. Representative methods use long-context training to improve models' utilization of extended conversation histories (Fu et al., 2024; An et al., 2024). These methods avoid additional memory structures but incur high input costs and context-utilization issues, including the "lost-in-the-middle" phenomenon (Shi et al., 2023; Liu et al., 2024; Laban et al., 2026). In contrast, external memory methods (Xu et al., 2022; Maharana et al., 2024; Li et al., 2025; Zhong et al., 2024; Xu et al., 2025; Pan et al., 2025; Wu et al., 2025) store interaction histories as non-parametric memories for on-demand access, typically through retrieval-augmented generation (RAG) (Lewis et al., 2020; Guu et al., 2020). Existing studies primarily focus on memory construction, updating, organization, and retrieval, including summarizing histories into memory records or user profiles (Chen et al., 2025b; Zhong et al., 2024), recursively updating memories through summarization or reflection (Wang et al., 2025; Tan et al., 2025), and organizing them as turns, sessions, semantic segments, or structured graph nodes (Wu et al., 2025; Pan et al., 2025; Xu et al., 2025). However, they typically treat historical user queries and LLM responses jointly, with limited attention to their distinct roles or fine-grained selection of retrieved LLM responses.

Context Selection and Compression. Context compression aims to remove redundant information, reduce noise, and lower downstream input cost. Existing methods can be broadly categorized into soft and hard compression. Soft compression encodes long texts into compact continuous representations, such as learned gist tokens, summary vectors, or memory slots (Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024), but typically requires additional training or model adaptation. Hard compression deletes, extracts, or rewrites text for plug-and-play use, including abstractive methods (Yoon et al., 2024; Xu et al., 2024) that generate concise summaries and extractive methods (Li et al., 2023; Jiang et al., 2023; 2024; Pan et al., 2024; Xu et al., 2024; Hwang et al., 2025) that select salient sentences. Methods can also be classified as query-aware or query-agnostic depending on whether they use the current query. Query-aware methods (Jiang et al., 2024; Pan et al., 2024; Xu et al., 2024; Creswell et al., 2023) retain information relevant to the current query, whereas query-agnostic methods (Jiang et al., 2023; Li et al., 2023) compress context based on textual importance, information entropy, or generic summarization, potentially discarding query-critical evidence. Overall, existing methods often treat conversational context as homogeneous text, with limited consideration of query-response roles or LLM-dependent utility. The closest work to ours (Huang et al., 2026) studies context pollution from prior LLM responses and classifies whether to keep or omit the full response history in single-session conversations. In contrast, we begin from the more fundamental query-response asymmetry in length and informational role. In retrieval-based long-term memory, retrieved turns from different sessions can contain historical LLM responses with substantially different utility for the current query and downstream LLM. Uniform retention or removal therefore overlooks this heterogeneity, motivating fine-grained, LLM-conditioned response selection.

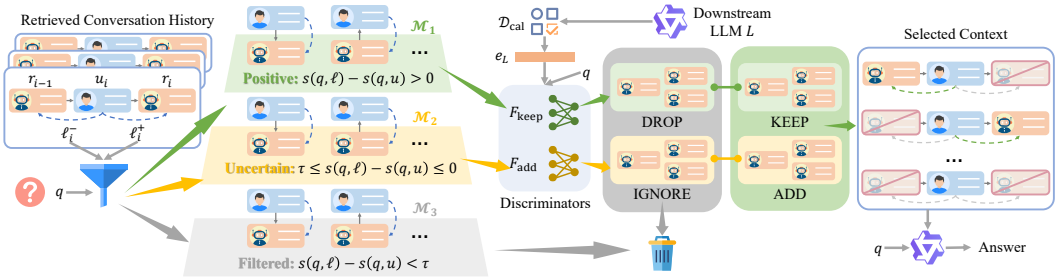


Figure 3: Overview of QARS. Historical user queries serve as memory anchors, with adjacent LLM responses treated as local candidates. Stage 1 filters LLM response candidates by semantic relevance, and Stage 2 selects the resulting candidates according to the needs of the downstream LLM.

3 Problem Formulation

We study post-retrieval context selection in long-term conversational memory. Given an interaction history H and a current query q, retrieval yields a memory context $\mathcal{M}_q$ from H to support a downstream LLM L in answering q, where historical LLM responses in $\mathcal{M}_q$ may have been generated by LLMs different from L. We focus on transforming $\mathcal{M}_q$ into a more compact context $\mathcal{M}'_q$ conditioned on q and L before generation to reduce input tokens while preserving answer performance.

Under turn-level retrieval, memories typically pair each historical user query with its following LLM response as (u_i, r_i). However, a user query may also depend on the preceding response, for example when it contains a follow-up, reference, or elliptical expression difficult to interpret in isolation. We therefore represent each retrieved user query with its adjacent LLM responses as a user-query-anchored candidate structure $x_i = (r_{i-1}, u_i, r_i)$. The historical user query u_i is always retained as the memory anchor, while adjacent LLM responses are treated as optional candidates. The same formulation extends to session-level retrieval, where each retrieved session comprises multiple such structures. Our task is to perform fine-grained selection over response candidates according to the current query q and downstream LLM L, producing $\mathcal{M}'_q$ by retaining only useful LLM responses.

4 Method

This section presents QARS, our user-query-anchored, downstream-LLM-aware framework for historical LLM response selection, as illustrated in Figure 3. For each retrieved user query u_i, we regard its adjacent LLM responses r_{i-1} and r_i as response candidates linked to the user query. Historical LLM response candidates vary substantially in how difficult their utility is to determine: many can be screened by semantic relevance alone, while the remaining candidates require finer-grained assessment of their utility to the downstream LLM. We therefore adopt a coarse-to-fine design to balance efficient candidate pruning with accurate LLM-specific selection: Stage 1 partitions candidates by semantic relevance into positive, uncertain, or filtered groups, while Stage 2 performs downstream-LLM-aware selection over the remaining candidates using retention and addition discriminators.

4.1 Semantic Relevance Filtering of Historical LLM Responses

The first stage filters historical LLM responses based on their semantic relevance to the current query. Rather than assessing whether an LLM response is relevant in isolation, we ask whether it adds useful information beyond the historical user query, which is already retained in the context. For example, an LLM response may be highly relevant to the current query yet largely repeat the user query evidence, while another may strengthen the connection between the user query and the current query by supplying complementary information. We therefore use the relevance of the user query as a baseline and measure how it changes after incorporating each LLM response candidate.

Local Semantic Links and Incremental Contribution. For each user query u_i, the preceding LLM response r_{i-1} provides local context for interpreting it, while the following LLM response r_i is generated from it. We denote r_{i-1} and r_i as r_i^- and r_i^+, and construct two local semantic links:

$$\ell_i^- = (r_i^-, u_i), \quad \ell_i^+ = (u_i, r_i^+). \tag{1}$$

Let $s(q, c) = \mathrm{sim}(E(q), E(c))$ denote the relevance between the current query q and context c, computed using a fixed text encoder E. Taking the relevance of the historical user query as the baseline, we measure the incremental semantic contribution of each LLM response candidate as:

$$\Delta_i^-(q) = s(q, \ell_i^-) - s(q, u_i), \qquad \Delta_i^+(q) = s(q, \ell_i^+) - s(q, u_i). \tag{2}$$

For $d \in \{-, +\}$, a larger $\Delta_i^d(q)$ indicates a greater increase in the historical user query's relevance to the current query after incorporating the corresponding LLM response.

Candidate Partitioning. We partition LLM response candidates by semantic relevance changes:

$$r_i^d \in \begin{cases} \mathcal{M}_1, & \Delta_i^d(q) > 0, \\ \mathcal{M}_2, & \tau \le \Delta_i^d(q) \le 0, \\ \mathcal{M}_3, & \Delta_i^d(q) < \tau. \end{cases} \tag{3}$$

Here, $\tau < 0$ is the filtering threshold. Candidates in $\mathcal{M}_1$ exhibit positive relevance changes, while those in $\mathcal{M}_2$ have uncertain relevance changes that cannot be reliably resolved by semantic relevance alone. Candidates in $\mathcal{M}_3$, which account for a substantial portion of the candidate space, clearly reduce relevance to the current query and are therefore filtered without downstream-LLM-specific judgment. We pass only $\mathcal{M}_1$ and $\mathcal{M}_2$ to the downstream-LLM-aware response selection, reserving LLM-specific selection for candidates whose necessity cannot be determined from semantic relevance alone. This substantially narrows the candidate space, reducing the cost of LLM-specific supervision and inference as well as the complexity of Stage 2.

4.2 Downstream-LLM-Aware Selection of Historical LLM Responses

The necessity of a historical LLM response can vary across downstream LLMs, even given the same semantic relevance. A candidate in $\mathcal{M}_1$ may provide clear semantic value yet be unnecessary if the downstream LLM can recover the information from the user query evidence, whereas a candidate in $\mathcal{M}_2$ may remain useful by providing complementary cues that the LLM cannot reliably derive or exploit from that evidence alone. Stage 2 therefore performs LLM-aware calibration over the two candidate groups, pruning unnecessary candidates from $\mathcal{M}_1$ and recovering useful ones from $\mathcal{M}_2$.

LLM-Conditioned Supervision. For each training query and downstream LLM, we simulate answer generation using the Stage 1 context $C^{(1)} = C_U \cup \mathcal{M}_1$, combining user query context C_U with positive candidates. If the downstream LLM answers correctly, we enumerate candidate subsets of $\mathcal{M}_1$ to identify a minimal sufficient subset that preserves correctness, labeling its candidates as KEEP and the remaining ones as DROP, while candidates in $\mathcal{M}_2$ are labeled IGNORE. Otherwise, we enumerate candidate subsets of $\mathcal{M}_2$ to identify a minimal supplementary subset that corrects the answer, labeling its candidates as ADD and the remaining ones as IGNORE, while candidates in $\mathcal{M}_1$ are labeled KEEP. Instances for which no such supplementary subset is found are discarded. These labels form two supervision sets, where y_i^d denotes the KEEP/DROP or ADD/IGNORE label:

$$\mathcal{D}_{\mathrm{keep}} = \{(L, q, u_i, r_i^d, y_i^d) \mid r_i^d \in \mathcal{M}_1\}, \quad \mathcal{D}_{\mathrm{add}} = \{(L, q, u_i, r_i^d, y_i^d) \mid r_i^d \in \mathcal{M}_2\}. \tag{4}$$

Further details of subset enumeration and label construction are provided in Appendix A.

Retention and Addition Discriminators. Based on the supervision above, we train a retention discriminator F_{keep} on $\mathcal{D}_{\mathrm{keep}}$ to determine whether positive candidates remain necessary for the downstream LLM, and an addition discriminator F_{add} on $\mathcal{D}_{\mathrm{add}}$ to determine whether uncertain candidates should be added. They share the same architecture but are trained independently with cross-entropy loss because they operate on different candidate distributions and objectives.

For either discriminator, the input combines an LLM representation e_L, which captures differences in how downstream LLMs utilize historical information, with the current query, historical user query, and LLM response candidate representations e_q, e_u, and e_r obtained using a fixed text encoder E. These representations are projected into a shared hidden space and concatenated:

$$z = \mathrm{Proj}_L(e_L) \oplus \mathrm{Proj}_q(e_q) \oplus \mathrm{Proj}_u(e_u) \oplus \mathrm{Proj}_r(e_r). \tag{5}$$

The concatenated representation is then processed by CrossLayers and a binary classification head:

$$h = \mathrm{CrossLayers}(z), \qquad \mathbf{p} = \mathrm{softmax}(\mathrm{Cls}(h)). \tag{6}$$

Here, $\oplus$ denotes concatenation, and $\mathbf{p}$ denotes the predicted probability distribution over the two classes, where the positive class corresponds to KEEP for F_{keep} and ADD for F_{add}. At inference, F_{keep} is applied to candidates in $\mathcal{M}_1$, while F_{add} is applied to those in $\mathcal{M}_2$; the predicted class determines whether each candidate is retained in or added to the final context.

4.3 BEHAVIOR-BASED LLM REPRESENTATION

To provide the discriminators with LLM-specific information, we derive the LLM representation e_L from answering behavior on a shared diagnostic calibration set. We probe each downstream LLM on the same query-context configurations and summarize its behavior into a compact representation, which serves as the LLM input to the discriminators.

Calibration Set. To construct a lightweight calibration set, we select queries that expose substantial behavioral differences among downstream LLMs. During supervision construction, we obtain the LLM-specific selected context for each training query, together with answer correctness before and after context calibration. We therefore evaluate each query using two quantitative indicators. *Context utilization divergence* is higher when different downstream LLMs select substantially different historical LLM responses for the same query, while *answer performance divergence* is higher when context calibration corrects the answers of some downstream LLMs but provides no such gain for others. We combine the rankings under these two indicators and select the top K queries. For each selected query q, we collect its Stage 1 context and distinct LLM-specific selected contexts, each paired with the gold answer, yielding the shared calibration set $\mathcal{D}_{\text{cal}} = \{(q_j, C_j, g_j)\}_{j=1}^{M}$, where C_j denotes either the Stage 1 context or an LLM-specific selected context, and M is the number of calibration instances. Details of the two indicators and ranking procedure are provided in Appendix B.

Behavior Representation Construction. We run each LLM L on all calibration instances and record its answer correctness as a behavior vector $\mathbf{b}_L = [s_{L,1}, s_{L,2}, \ldots, s_{L,M}]$, where $s_{L,j} \in \{0, 1\}$ indicates whether the j-th instance is answered correctly. To reduce the influence of instance difficulty on the behavior representation, we compute an instance-wise baseline $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_M]$, where $\mu_j = \frac{1}{N}\sum_{n=1}^{N} s_{n,j}$ is the average correctness of the N training LLMs on the j-th instance.

We then center the behavior vector by this baseline and append the LLM's overall calibration accuracy, $\text{Acc}_L = \frac{1}{M}\sum_{j=1}^{M} s_{L,j}$, to capture its overall performance:

$$e_L = [\mathbf{b}_L - \boldsymbol{\mu}; \text{Acc}_L] \in \mathbb{R}^{M+1}. \tag{7}$$

The resulting e_L is used as the LLM-specific conditioning input to both discriminators. For an unseen LLM, the same representation can be obtained by running it once on the fixed calibration set using the training-LLM baseline $\boldsymbol{\mu}$, enabling direct adaptation without retraining the discriminators.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Dataset. We use **ConvoMem** (CM) (Pakhomov et al., 2025) and **HaluMem** (HM) (Chen et al., 2025a) for training and evaluation, and **LongMemEval-S** (LME) (Wu et al., 2025) for out-of-domain evaluation. We report answer accuracy and the average input tokens per query as the metrics.

Downstream LLMs. We use six training-visible downstream LLMs: **Qwen3.5-9B**, **Qwen3.5-27B**, **Qwen3.5-122B-A10B** (Qwen Team, 2026), **Gemma-4-26B-A4B-it** (Team et al., 2026), **GLM-5.2** (Zeng et al., 2026), and **DeepSeek-V4-Flash** (Xu et al., 2026). Generalization is evaluated on two unseen LLMs: **Llama-3.3-70B-Instruct** (Grattafiori et al., 2024) and **GPT-5.1** (Singh et al., 2025). For brevity, we shorten Qwen3.5-122B-A10B, Gemma-4-26B-A4B-it, and Llama-3.3-70B-Instruct to Qwen3.5-122B, Gemma-4-26B, and Llama3.3-70B in tables and figures.

Baselines. We compare our method with three groups of baselines. First, retrieval-based conversational memory baselines include (1) **turn-level RAG**, (2) **session-level RAG**, (3) the segment-level memory method **SeCom** (Pan et al., 2025), and (4) the recursive summarization method **RSum** (Wang et al., 2025). Second, context-compression baselines include (5) **CompAct** (Yoon et al., 2024), (6) **RECOMP** (Xu et al., 2024), (7) **LLMLingua2** (Pan et al., 2024), (8) **EXIT** (Hwang et al., 2025). Third, we include reference baselines: (9) **Memory-Free**, which uses only the current query without memory; (10) **Response-Free**, which retains retrieved user queries while removing all historical LLM responses; (11) **Binary Oracle**, which retrospectively selects the better of RAG and Response-Free for each query, providing an upper bound for global keep-or-remove strategies (Huang et al., 2026); and (12) **Oracle**, which applies the bounded local search used for label construction to each test instance as an analysis-only upper bound. Our method and all context-compression baselines are evaluated with both turn-level and session-level retrieval inputs.

Table 1: Main results across six downstream LLMs on CM and HM. Acc. denotes LLM-judged answer accuracy (%), and Tok. denotes average input tokens per query, using the Qwen3.5 tokenizer. **Bold** and underlined values indicate the best and second-best Acc., respectively. T, S, SE, and M denote turn, session, segment, and summary granularity. DS-V4-Flash denotes DeepSeek-V4-Flash. *Binary Oracle is an oracle upper bound for the global keep-or-remove strategy (Huang et al., 2026).

Method	Gran.	Qwen3.5-9B				Qwen3.5-27B				Qwen3.5-122B			
		ConvoMem		HaluMem		ConvoMem		HaluMem		ConvoMem		HaluMem	
		Acc.↑	Tok.↓	Acc.↑	Tok.↓	Acc.↑	Tok.↓	Acc.↑	Tok.↓	Acc.↑	Tok.↓	Acc.↑	Tok.↓
Oracle	T	81.8	604	72.6	892	82.0	599	72.1	890	80.9	600	73.1	889
Binary Oracle* (Huang et al., 2026)	T	71.5	616	68.0	949	73.6	616	68.9	951	73.1	618	69.5	955
Memory-Free	–	31.1	42	24.2	38	32.0	42	19.4	38	33.0	42	28.0	38
Response-Free	T	62.6	579	59.4	874	64.7	579	60.4	874	63.9	579	60.5	874
	S	72.6	8070	69.9	7915	74.9	8070	73.7	7915	75.5	8070	72.4	7915
Retrieval-Based Methods													
RAG	T	64.8	975	60.4	1721	67.6	975	63.9	1721	67.1	975	64.4	1721
	S	75.8	15557	72.4	27624	78.8	15557	76.5	27624	**79.7**	15557	74.8	27624
SeCom (Pan et al., 2025)	SE	75.1	5169	71.3	24187	79.2	5169	75.3	24187	78.5	5169	75.5	24187
RSum (Wang et al., 2025)	M	46.2	538	34.0	711	54.4	538	42.4	711	48.5	538	39.1	711
Context-Compression Methods													
CompAct (Yoon et al., 2024)	T	55.0	562	50.1	917	61.3	562	52.3	917	57.7	562	53.2	917
	S	59.8	907	51.6	1309	64.9	907	53.8	1309	64.5	907	54.7	1309
RECOMP (Xu et al., 2024)	T	61.9	749	59.7	1357	68.0	749	62.9	1357	66.0	749	65.0	1357
	S	72.6	4306	69.8	5955	78.4	4306	72.6	5955	75.6	4306	72.2	5955
LLMLingua2 (Pan et al., 2024)	T	51.7	358	53.3	777	57.2	358	58.5	777	57.1	358	58.5	777
	S	72.5	3937	71.3	7504	75.9	3937	76.5	7504	73.3	3937	73.7	7504
EXIT (Hwang et al., 2025)	T	63.4	478	58.2	791	65.7	478	59.7	791	65.1	478	59.2	791
	S	54.8	530	49.7	1075	59.8	530	56.5	1075	60.1	530	57.0	1075
QARS (Ours)	T	70.5	617	64.1	890	73.6	627	66.6	896	73.4	655	68.1	901
	S	**79.1**	8198	**75.4**	7940	**79.8**	8316	**78.3**	7950	**79.7**	8446	**78.7**	7958

Method	Gran.	Gemma-4-26B				GLM-5.2				DS-V4-Flash			
		ConvoMem		HaluMem		ConvoMem		HaluMem		ConvoMem		HaluMem	
		Acc.↑	Tok.↓	Acc.↑	Tok.↓	Acc.↑	Tok.↓	Acc.↑	Tok.↓	Acc.↑	Tok.↓	Acc.↑	Tok.↓
Oracle	T	81.5	621	73.9	893	83.3	591	76.6	887	80.1	648	75.8	897
Binary Oracle* (Huang et al., 2026)	T	74.1	624	70.4	944	76.4	619	71.5	942	67.5	658	69.1	948
Memory-Free	–	30.0	42	31.7	38	30.2	42	20.6	38	27.7	42	9.6	38
Response-Free	T	65.7	579	63.1	874	67.0	579	63.9	874	58.0	579	60.8	874
	S	77.1	8070	74.9	7915	78.3	8070	72.1	7915	58.2	8070	62.6	7915
Retrieval-Based Methods													
RAG	T	69.7	975	64.6	1721	71.3	975	65.2	1721	60.9	975	62.1	1721
	S	78.8	15557	77.5	27624	80.3	15557	76.5	27624	62.1	15557	64.3	27624
SeCom (Pan et al., 2025)	SE	79.7	5169	77.5	24187	80.5	5169	75.6	24187	64.8	5169	63.7	24187
RSum (Wang et al., 2025)	M	44.6	538	37.8	711	46.1	538	40.2	711	38.7	538	40.3	711
Context-Compression Methods													
CompAct (Yoon et al., 2024)	T	59.8	562	54.6	917	59.3	562	54.1	917	53.1	562	49.5	917
	S	68.0	907	60.7	1309	66.4	907	59.8	1309	59.4	907	56.4	1309
RECOMP (Xu et al., 2024)	T	68.1	749	64.6	1357	68.1	749	63.7	1357	57.7	749	62.2	1357
	S	77.5	4306	74.4	5955	80.9	4306	72.8	5955	60.7	4306	62.4	5955
LLMLingua2 (Pan et al., 2024)	T	58.2	358	61.8	777	56.2	358	61.8	777	47.7	358	47.9	777
	S	76.9	3937	76.0	7504	74.7	3937	74.8	7504	59.8	3937	59.1	7504
EXIT (Hwang et al., 2025)	T	65.7	478	61.0	791	64.9	478	63.5	791	60.5	478	57.1	791
	S	60.2	530	54.8	1075	60.6	530	54.6	1075	55.5	530	50.4	1075
QARS (Ours)	T	73.9	621	68.5	890	75.9	593	70.0	891	66.5	746	65.0	893
	S	**82.4**	8207	**80.1**	7942	**84.0**	8163	**79.3**	7947	**68.9**	8990	**70.5**	7979

Implementation Details. QARS operates as a post-retrieval selection module and can be integrated into existing RAG pipelines without modifying the retriever or downstream LLM. We use BGE-M3 (Chen et al., 2024) as the retriever and fixed text encoder. All methods retrieve the top $k = 10$ memory units. In Stage 1, we set the threshold $\tau = -0.05$. For calibration, we select the top $K = 100$ queries, yielding $M = 438$ unique query-context instances. Both discriminators are trained for 50 epochs using AdamW (Loshchilov & Hutter, 2019) with a learning rate of 1e-4 and a batch size of 1,024. Training uses a single NVIDIA GeForce RTX 5090 GPU. Online selection takes 0.45 s/query after retrieval; offline label construction takes 62.7 min per LLM-dataset pair and calibration 3.86 min per LLM. Detailed efficiency analysis is provided in Appendix H.

5.2 Main Results

QARS outperforms retrieval and compression baselines. As shown in Table 1, averaged over six downstream LLMs, under turn-level retrieval, our method improves accuracy over RAG by 5.4% and 3.6% on CM and HM, while reducing input tokens by 34.0% and 48.1%, respectively. It also outperforms all compression baselines by at least 7.3% and 4.0% on CM and HM, respectively. Notably, our method is only 1.5% below Binary Oracle, the upper bound of the global keep-or-remove strategy (Huang et al., 2026), while still using 2.3% fewer input tokens.

Table 2: Cross-dataset generalization on LME under turn-level retrieval across six downstream LLMs.

Method	Qwen3.5-9B		Qwen3.5-27B		Qwen3.5-122B	
	Acc.↑	Tok.↓	Acc.↑	Tok.↓	Acc.↑	Tok.↓
Response-Free	73.2	1128	77.6	1128	77.0	1128
RAG	79.0	5340	81.8	5340	82.2	5340
RECOMP	75.2	1714	73.6	1714	76.4	1714
LLMLingua2	59.2	1896	64.4	1896	65.4	1896
EXIT	78.8	966	77.8	966	78.2	966
QARS (Ours)	**84.0**	1335	**83.2**	1337	**84.2**	1549

Method	Gemma-4-26B		GLM-5.2		DS-V4-Flash	
	Acc.↑	Tok.↓	Acc.↑	Tok.↓	Acc.↑	Tok.↓
Response-Free	80.8	1128	80.4	1128	70.8	1128
RAG	82.2	5340	82.4	5340	78.4	5340
RECOMP	77.8	1714	77.8	1714	66.0	1714
LLMLingua2	67.4	1896	69.6	1896	54.8	1896
EXIT	81.2	966	81.4	966	74.6	966
QARS (Ours)	**85.8**	1286	**86.6**	1246	**82.8**	1829

Table 3: Results on unseen LLMs under turn-level retrieval on CM and LME.

Method	Llama3.3-70B		GPT-5.1	
	Acc.↑	Tok.↓	Acc.↑	Tok.↓
ConvoMem				
Response-Free	58.3	579	75.5	579
RAG	65.2	975	78.4	975
RECOMP	64.5	749	73.0	749
LLMLingua2	54.7	358	66.4	358
EXIT	60.9	478	70.4	478
QARS (Ours)	**68.6**	637	**80.6**	603
LongMemEval				
Response-Free	72.2	1128	84.0	1128
RAG	73.6	5340	86.2	5340
RECOMP	72.8	1714	80.4	1714
LLMLingua2	58.0	1896	77.0	1896
EXIT	76.6	966	81.8	966
QARS (Ours)	**80.8**	1321	**87.6**	1243

QARS remains effective across retrieval granularities. The gains persist when retrieval shifts from turns to sessions. Under session-level retrieval, our method improves average accuracy over RAG by 3.2% while reducing input tokens by 62.2%, and outperforms SeCom by 3.3% with 44.3% fewer tokens. These results demonstrate robust effectiveness across different retrieval granularities.

QARS achieves consistent gains across downstream LLMs while constructing LLM-specific contexts. As shown in Table 1, averaged across datasets, it improves over RAG for all six training-visible downstream LLMs by 1.4-6.5% under turn- and session-level retrieval. Unlike LLM-agnostic baselines that use identical contexts across LLMs, our method selects contexts of different lengths across both granularities according to their needs for historical LLM responses.

5.3 Generalization Results

Generalization to an unseen dataset. As shown in Table 2, on LME, which is excluded from training, QARS improves average accuracy over RAG by 3.4% while reducing input tokens by 73.2%, and consistently outperforms compression baselines, demonstrating cross-dataset generalization.

Generalization to unseen downstream LLMs. As shown in Table 3, for the unseen LLMs Llama-3.3-70B and GPT-5.1, our method consistently outperforms RAG across CM and LME, improving accuracy by 1.4-7.2% while reducing input tokens by 34.7-76.7%. It also outperforms all compression baselines, demonstrating transfer to these unseen LLMs without retraining the discriminators.

5.4 Analysis of Historical LLM Response Selection

To understand where the accuracy gains and context reduction of our method come from, we categorize queries according to the effect of historical LLM responses by comparing turn-level RAG with Response-Free. We distinguish four cases: **Both Correct**, where both configurations answer correctly; **Negative**, where only Response-Free is correct, indicating that the historical LLM responses are harmful; **Positive**, where only RAG is correct, indicating that they are beneficial; and **Both Wrong**, where neither configuration answers correctly. These categories capture whether historical LLM responses are unnecessary, harmful, beneficial, or insufficient. Figure 4 reports their proportions with the accuracy and relative number of historical LLM responses retained by our method and Oracle, normalized by turn-level RAG.

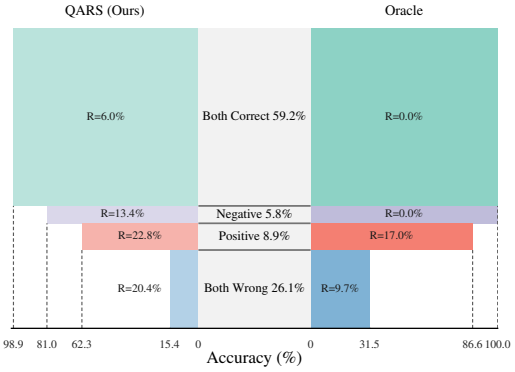


Figure 4: Category-wise comparison of QARS and Oracle. Band height indicates category proportion, and R denotes the ratio of selected response count to that of turn-level RAG. Results are aggregated over three datasets and six LLMs.

Oracle reveals substantial room for selective historical LLM response retention. Overall, Oracle reaches 81.0% accuracy, 12.9% above turn-level RAG, while selecting only 4.0% as many responses. This headroom is category-dependent: Both Correct and Negative queries account for 65.0% of cases and require no historical LLM responses under Oracle, whereas Positive and Both Wrong queries retain only 17.0% and 9.7% as many responses as RAG. These results show that most historical LLM responses can be removed, with only sparse responses remaining useful.

QARS exploits this optimization space by selectively retaining historical LLM responses based on their usefulness. For Both Correct and Negative queries, it selects only 6.0% and 13.4% as many responses as RAG while achieving 98.9% and 81.0% accuracy. For Positive and Both Wrong queries, retention rises to 22.8% and 20.4%, with accuracies of 62.3% and 15.4%. This contrast supports historical user queries as the primary memory basis, with LLM responses retained when they provide complementary information. Thus, context reduction and accuracy gains arise from suppressing redundant or harmful responses while retaining useful ones.

QARS approximates the Oracle's selection pattern. Both retain fewer historical LLM responses for Both Correct and Negative queries than for Positive and Both Wrong queries, suppressing unnecessary or harmful responses while preserving useful ones. Our method and Oracle select 11.7% and 4.0% as many responses as RAG, with 72.8% versus 81.0% accuracy. The smallest gap occurs for Both Correct (98.9% vs. 100.0%), while accuracy gaps remain in the other categories.

5.5 Component and Architecture Ablations

Ablation of the Two Stages. As shown in Table 4, the two stages progressively narrow the retrieved context. Stage 1 filters by semantic relevance, while Stage 2 resolves remaining ambiguity for the downstream LLM. Stage 1 partitions 4.0%, 39.0%, and 57.0% of candidates into $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$, respectively, reducing the candidate space. Removing both stages lowers accuracy by 4.4% and increases input tokens. Stage 1 alone yields short contexts but a 7.3% accuracy drop, showing that semantic relevance alone cannot determine candidate utility. With Stage 2 alone, all candidates are treated as uncertain, resulting in a 2.9% accuracy drop and longer contexts. This shows semantic pre-filtering and LLM-aware selection are complementary rather than redundant.

Table 4: Ablation of the two-stage framework on CM and LME. $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ report the proportions (%) of candidates in the three Stage 1 categories.

Stage 1	Stage 2	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	ConvoMem		LongMemEval	
					Acc.↑	Tok.↓	Acc.↑	Tok.↓
✗	✗	–	–	–	66.9	975	81.0	5340
✓	✗	–	–	–	62.5	623	79.7	1263
✗	✓	0.0%	100.0%	0.0%	69.2	732	81.8	2212
✓	✓	4.0%	39.0%	57.0%	72.3	643	84.4	1430

Ablation of LLM Representations. As shown in Figure 5, removing the LLM representation reduces average accuracy by 4.4% on CM, with consistent degradation across all six downstream LLMs. Without it, all LLMs share the same context. In contrast, the behavior-based representation enables LLM-specific context selection by adapting the amount and composition of historical LLM responses to each LLM's use of historical information, consistently improving accuracy.

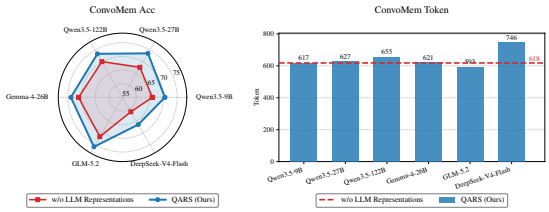


Figure 5: Effect of LLM representations on CM. Left: Accuracy across LLMs. Right: selected context length with and without LLM representations.

6 Conclusion

In this work, we show that historical user queries and LLM responses play distinct roles in long-term conversational memory: user queries are concise and information-dense memory anchors, while LLM responses are substantially longer, largely compressible, and conditionally useful. Based on this asymmetry, QARS selectively retains historical LLM responses according to current-query relevance and downstream-LLM needs, maintaining or improving answer accuracy while reducing input tokens across experimental settings. More broadly, these findings motivate a shift from uniformly preserving retrieved conversation histories toward memory systems that explicitly distinguish the roles of different interaction components and adapt context construction to the downstream LLM.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript editing, including language refinement, readability improvement, grammar checking, and sentence rephrasing. Generative AI tools were not used for research ideation, methodology development, experimental design, or interpretation of results. We take full responsibility for the content of the manuscript.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of our method in Section 4 and Appendices A and B. Experimental settings and implementation details are provided in Section 5.1 and Appendix D. All datasets and LLMs used in this work are publicly available. Our source code will be released upon publication to facilitate replication.

REFERENCES

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. Make your llm fully utilize the context. *Advances in Neural Information Processing Systems*, 37:62160–62188, 2024.

Ding Chen, Simin Niu, Kehang Li, Peng Liu, Xiangping Zheng, Bo Tang, Xinchi Li, Feiyu Xiong, and Zhiyu Li. Halumem: Evaluating hallucinations in memory systems of agents. *arXiv preprint arXiv:2511.03506*, 2025a.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the association for computational linguistics: ACL 2024*, pp. 2318–2335, 2024.

Nuo Chen, Hongguang Li, Jianhui Chang, Juhua Huang, Baoyuan Wang, and Jia Li. Compress to impress: Unleashing the potential of compressive memory in real-world long-term conversations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 755–773, 2025b.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846, 2023.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3Pf3Wg6o-A4.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 14125–14134, 2024.

Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uREj4ZuGJE.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Jenny Y. Huang, Leshem Choshen, Wei Sun, Omar Khattab, Ramón Fernandez Astudillo, Mehul Damani, Tamara Broderick, and Jacob Andreas. Do LLMs benefit from their own words? In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=l6T6MrLgWp.

Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C Park. Exit: Context-aware extractive compression for enhancing retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4895–4924, 2025.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VKGTGGcwl6.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. Hello again! llm-powered personalized agent for long-term dialogue. 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. 5259–5276, 2025.

Xiaopeng Li, Pengyue Jia, Derong Xu, Yi Wen, Yingyi Zhang, Wenlin Zhang, Wanyu Wang, Yichao Wang, Zhaocheng Du, Xiangyang Li, et al. A survey of personalization: From rag to agent. *ACM Transactions on Information Systems*, 44(4):1–39, 2026.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*, 2024.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 6342–6353, 2023.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

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.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352, 2023.

Egor Pakhomov, Erik Nijkamp, and Caiming Xiong. Convomem benchmark: Why your first 150 conversations don't need rag. *arXiv preprint arXiv:2511.10523*, 2025.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, 2024.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. Secom: On memory construction and retrieval for personalized conversational agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xKDZAW0He3.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2463–2473, 2019.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pp. 31210–31227. PMLR, 2023.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33: 16857–16867, 2020.

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.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, 639:130193, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with context compression and selective augmentation. In *International Conference on Learning Representations*, volume 2024, pp. 43478–43502, 2024.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 5180–5197, 2022.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=FiM0M8gcct.

Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. Compact: Compressing retrieved documents actively for question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21424–21439, 2024.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. Do LLMs recognize your preferences? evaluating personalized preference following in LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QWunLKbBGF.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

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.

APPENDIX CONTENTS

A Design Details of LLM-Conditioned Label Construction **15**

B Design Details of Calibration Set Construction **16**

C Extended Design for Session-Level Retrieval Units **17**

D Experimental Details **17**

D.1 Dataset Details 17

D.2 Evaluation Details 18

D.3 Downstream LLM Deployment 18

D.4 Baseline Details 19

E Additional Ablations and Sensitivity to Key Settings **19**

E.1 Discriminator Design Ablation 19

E.2 Effect of the Semantic Relevance Filtering Threshold τ 19

E.3 Performance across Retrieval Depths k 20

E.4 Performance across Different Retrievers 20

F Calibration Set Analysis **20**

F.1 Discriminative Capacity of the Diagnostic Calibration Set 20

F.2 Calibration Set Size and Selection Strategy 21

F.3 Ablation of Diagnostic Query Selection Indicators 21

G LLM Representation and Specificity Analysis **22**

G.1 Comparison with ID-Based Representation 22

G.2 LLM-Specificity Analysis 22

H Efficiency Analysis **23**

I Case Study **24**

A Design Details of LLM-Conditioned Label Construction

As described in Section 4.2, Stage 1 retains the user query context C_U and partitions historical LLM response candidates into $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$. Since $\mathcal{M}_3$ is filtered in Stage 1, we construct LLM-conditioned supervision only for candidates in $\mathcal{M}_1$ and $\mathcal{M}_2$. For each training query and downstream LLM, we assign a label to each historical LLM response candidate according to whether it is necessary for preserving or correcting the LLM's answer. Algorithm 1 summarizes the procedure.

Algorithm 1 LLM-Conditioned Label Construction

Require: User query context C_U, historical LLM response candidate sets $\mathcal{M}_1, \mathcal{M}_2$, query q, gold answer g, downstream LLM L, judge function J, addition budget B
Ensure: A label for each LLM response candidate in $\mathcal{M}_1, \mathcal{M}_2$, or discard the query-LLM instance

$C^{(1)} \leftarrow C_U \cup \mathcal{M}_1$ ▷ Stage 1 context
if $J(L(q, C^{(1)}), g) = 1$ **then**
  for $m = 0, \ldots, |\mathcal{M}_1|$ **do** ▷ Search for a minimum sufficient subset
    Evaluate subsets $\mathcal{K} \subseteq \mathcal{M}_1$ with $|\mathcal{K}| = m$
    if a subset $\mathcal{K}$ satisfies $J(L(q, C_U \cup \mathcal{K}), g) = 1$ **then**
      $\mathcal{K}^\star \leftarrow \mathcal{K}$; **break**
    end if
  end for
  Assign KEEP labels to candidates in $\mathcal{K}^\star$ and DROP labels to candidates in $\mathcal{M}_1 \setminus \mathcal{K}^\star$
  Assign IGNORE labels to candidates in $\mathcal{M}_2$
else
  for $m = 1, \ldots, \min(B, |\mathcal{M}_2|)$ **do** ▷ Search for a minimum supplementary subset
    Evaluate subsets $\mathcal{P} \subseteq \mathcal{M}_2$ with $|\mathcal{P}| = m$
    if a subset $\mathcal{P}$ satisfies $J(L(q, C^{(1)} \cup \mathcal{P}), g) = 1$ **then**
      $\mathcal{P}^\star \leftarrow \mathcal{P}$; **break**
    end if
  end for
  if $\mathcal{P}^\star$ exists **then**
    Assign ADD labels to candidates in $\mathcal{P}^\star$ and IGNORE labels to candidates in $\mathcal{M}_2 \setminus \mathcal{P}^\star$
    Assign KEEP labels to candidates in $\mathcal{M}_1$
  else
    Discard the query-LLM instance
  end if
end if

Retention Label Construction. The Stage 1 context $C^{(1)}$ contains all positive historical LLM response candidates in $\mathcal{M}_1$. If the downstream LLM answers correctly under $C^{(1)}$, these candidates are collectively sufficient, although some may be unnecessary for that LLM. We therefore search for a minimum-cardinality subset of $\mathcal{M}_1$ that preserves answer correctness, evaluating subsets in increasing order of size until a sufficient one is found. Candidates in this subset are labeled KEEP, while the remaining candidates in $\mathcal{M}_1$ are labeled DROP. Since a correct answer is obtained without candidates from $\mathcal{M}_2$, they are labeled IGNORE.

Addition Label Construction. If the downstream LLM answers incorrectly under the Stage 1 context, the current context does not yield sufficient support for a correct answer. We therefore examine whether candidates in $\mathcal{M}_2$, despite their uncertain semantic contribution, can provide complementary information or useful cues that help the LLM focus, organize, or reason over the available evidence. We search for a minimum-cardinality supplementary subset of $\mathcal{M}_2$, evaluating candidate subsets in increasing order of size until a successful one is found. Unlike retention search, which is guaranteed to have a feasible solution because the full $\mathcal{M}_1$ already yields a correct answer, addition search may have no feasible solution. Moreover, $\mathcal{M}_2$ typically contains more candidates than $\mathcal{M}_1$, resulting in a substantially larger search space. We therefore restrict the supplementary subset to at most B candidates. We set $B = 2$ in all experiments. Candidates in the successful subset are labeled ADD, while the remaining candidates in $\mathcal{M}_2$ are labeled IGNORE. Candidates in $\mathcal{M}_1$ are labeled KEEP, as they remain in the Stage 1 context during the addition search. If no successful supplementary subset is found within the budget, the corresponding query-LLM instance is discarded.

Candidate-Level Supervision. The constructed labels provide candidate-level supervision for the two discriminators. In $\mathcal{D}_{\text{keep}}$, $y_i^d = 1$ and 0 correspond to KEEP and DROP, respectively; in $\mathcal{D}_{\text{add}}$, they correspond to ADD and IGNORE, respectively. Thus, each supervision instance reflects the utility of an individual historical LLM response candidate for a specific query-LLM pair.

Search Scope and Label Interpretation. Historical LLM response candidates cannot always be evaluated independently, as their utility may be affected by complex combinatorial effects such as complementarity, redundancy, substitution, and interference. Ideally, label construction would jointly consider all possible combinations of candidates in $\mathcal{M}_1$ and $\mathcal{M}_2$, but exhaustively searching this space is computationally prohibitive. Given the resulting combinatorial search space, we follow the search procedure described above, yielding a local rather than globally optimal solution. A more exhaustive joint search could in principle produce better supervision, but at a substantially higher construction cost. Empirically, however, the supervision obtained from our restricted search already enables the two discriminators to achieve strong downstream performance, indicating that it provides an effective practical approximation without requiring exhaustive combinatorial search.

B DESIGN DETAILS OF CALIBRATION SET CONSTRUCTION

As introduced in Section 4.3, our behavior-based LLM representation is constructed from a shared diagnostic calibration set. This section details how the calibration set is constructed. The goal is to obtain a compact set of queries that can effectively distinguish how different downstream LLMs utilize historical information. Rather than uniformly sampling training queries, we leverage the behavioral signals obtained during LLM-conditioned supervision construction and prioritize queries that exhibit substantial cross-LLM differences. Specifically, we characterize each query from two complementary perspectives: differences in the historical LLM responses selected by different downstream LLMs and differences in the answer-performance gains induced by downstream-LLM-aware response selection.

Context Utilization Divergence. A query is more informative for distinguishing LLM behaviors when different downstream LLMs require different historical LLM responses. Therefore, we measure the divergence between LLMs' selected response candidates for the same query. For a query q, let $\mathcal{S}_n(q)$ denote the set of LLM response candidates selected for downstream LLM L_n during LLM-conditioned supervision construction. For any two training LLMs L_i and L_j, we measure the difference between their selected historical LLM responses using the Jaccard distance:

$$d_J(\mathcal{S}_i(q), \mathcal{S}_j(q)) = 1 - \frac{|\mathcal{S}_i(q) \cap \mathcal{S}_j(q)|}{|\mathcal{S}_i(q) \cup \mathcal{S}_j(q)|}. \tag{8}$$

When both sets are empty, we define their Jaccard distance as 0. Given the training LLM set $\mathcal{L} = \{L_n\}_{n=1}^N$, the context utilization divergence is defined as:

$$\text{CtxDis}(q) = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} d_J(\mathcal{S}_i(q), \mathcal{S}_j(q)). \tag{9}$$

A higher $\text{CtxDis}(q)$ indicates that LLMs exhibit more diverse context selection behaviors for the query, making it more informative for characterizing heterogeneous context-utilization patterns.

Answer Performance Divergence. A query is more informative for distinguishing LLM behaviors when downstream-LLM-aware response selection benefits only a subset of LLMs. Therefore, we measure whether response selection improves the answers of some LLMs but not others. Let $s_n^{(1)}(q) \in \{0,1\}$ and $s_n^{(2)}(q) \in \{0,1\}$ denote the correctness of downstream LLM L_n under the Stage 1 context and its LLM-specific selected context, respectively. We define a selection-benefit indicator as:

$$b_n(q) = \mathbb{I}[s_n^{(1)}(q) = 0 \wedge s_n^{(2)}(q) = 1]. \tag{10}$$

The numbers of LLMs that benefit and do not benefit from response selection are:

$$n_{\text{gain}}(q) = \sum_{n=1}^{N} b_n(q), \qquad n_{\text{no-gain}}(q) = N - n_{\text{gain}}(q). \tag{11}$$

We define the answer performance divergence of query q as:

$$\text{GainDis}(q) = n_{\text{gain}}(q) \cdot n_{\text{no-gain}}(q). \tag{12}$$

A higher $\mathrm{GainDis}(q)$ indicates that response selection improves the answers of some LLMs but not others, revealing heterogeneous dependence on historical LLM responses across LLMs.

Query Selection and Instance Construction. We rank the training queries in descending order according to $\mathrm{CtxDis}(q)$ and $\mathrm{GainDis}(q)$, obtaining $\mathrm{rank}_{\mathrm{Ctx}}(q)$ and $\mathrm{rank}_{\mathrm{Gain}}(q)$, respectively. We combine the two rankings as:

$$\mathrm{rank}(q) = \mathrm{rank}_{\mathrm{Ctx}}(q) + \mathrm{rank}_{\mathrm{Gain}}(q). \tag{13}$$

Queries with smaller combined ranks exhibit stronger cross-LLM differences under the two diagnostic indicators. We select the top K queries according to this combined ranking to form the diagnostic query set $\mathcal{Q}_{\mathrm{cal}}$. For each $q \in \mathcal{Q}_{\mathrm{cal}}$, we collect its Stage 1 context $C_q^{(1)}$ and the LLM-specific selected context $C_{q,n}^{(2)}$ obtained for each training LLM L_n. Each context is paired with the query and its gold answer g_q, yielding:

$$\mathcal{D}_{\mathrm{cal}} = \{(q, C_q^{(1)}, g_q) \mid q \in \mathcal{Q}_{\mathrm{cal}}\} \cup \{(q, C_{q,n}^{(2)}, g_q) \mid q \in \mathcal{Q}_{\mathrm{cal}}, L_n \in \mathcal{L}\}. \tag{14}$$

Including both the shared Stage 1 context and LLM-specific selected contexts exposes LLMs to different context configurations for the same queries, providing richer behavioral observations for constructing their representations. Duplicate query-context pairs are retained only once. Let the resulting calibration set contain M instances, written as $\mathcal{D}_{\mathrm{cal}} = \{(q_j, C_j, g_j)\}_{j=1}^{M}$.

C Extended Design for Session-Level Retrieval Units

QARS can be directly extended to retrieval units containing multiple interaction turns, such as sessions. Given a retrieval unit with T turns, $S = [u_1, r_1, u_2, r_2, \ldots, u_T, r_T]$, we treat each historical user query as an anchor and construct local candidate structures $x_i = (r_{i-1}, u_i, r_i)$, where $r_0 = \varnothing$. A historical LLM response r_i can be associated with both u_i and u_{i+1}, serving as the following response to the former and the preceding response to the latter. We therefore process the two local relationships (u_i, r_i) and (r_i, u_{i+1}) independently through the same two-stage procedure, and average all available selection probabilities to determine whether r_i is retained. This local decomposition allows the trained discriminators to be directly applied to coarser retrieval units, such as sessions, without additional training.

D Experimental Details

D.1 Dataset Details

Table 5: Statistics of the datasets and experimental splits. # Disc. Inst. denotes the number of candidate-level discriminator training instances constructed from the training queries.

Dataset	Training		Test		
	# Queries	# Disc. Inst.	# Queries	Avg. Sessions	Avg. Turns
ConvoMem (CM)	2000	91152	2000	100.0	1943.0
HaluMem (HM)	2000	37182	821	68.3	1485.3
LongMemEval (LME)	–	–	500	47.7	246.8
Overall	4000	128334	3321	84.3	1574.5

We use three long-term conversational memory question-answering datasets: ConvoMem (Pakhomov et al., 2025), HaluMem (Chen et al., 2025a), and LongMemEval (Wu et al., 2025). These datasets feature long, multi-session, user-centric interaction histories paired with queries whose supporting evidence may be distributed across multiple sessions. The queries cover diverse information accumulated during user-LLM interactions, including personalized facts and preferences, information updates, and temporal reasoning. For ConvoMem, we adopt the 100-session split. For LongMemEval, we use the LongMemEval-S split.

ConvoMem and HaluMem are used for both training and evaluation. For ConvoMem, we randomly sample 2,000 queries for training and another 2,000 queries for testing. For HaluMem, we randomly

sample 2,000 queries for training and use all remaining 821 available queries for testing. In both datasets, we ensure that the conversation histories associated with training and test queries do not overlap. The 4,000 training queries yield 128,334 candidate-level labeled instances for discriminator training. LongMemEval contains 500 queries, all of which are used exclusively for evaluation. Overall, we evaluate on 3,321 test queries, with LongMemEval additionally serving to assess cross-dataset generalization. Detailed dataset statistics are provided in Table 5.

D.2 Evaluation Details

We evaluate answer correctness using an LLM-as-a-Judge (Zheng et al., 2023) protocol. Consistent with the evaluation objectives of these datasets, performance is measured by the correctness of the final downstream answer. For LongMemEval, we adopt its official evaluation prompt. For ConvoMem and HaluMem, we use the prompt shown in Figure 6. Qwen3-Max serves as the judge model in all experiments. All methods are evaluated using the same prompts, judge model, and evaluation procedure to ensure a fair comparison.

Prompt for LLM Judge

I will give you a question, a correct answer, and a response from a model. Please determine whether the model response is correct based only on the provided information.

Answer yes if the model response gives the correct answer, is semantically equivalent to the correct answer, or clearly entails the correct answer.

Answer no if the model response is incorrect, contradicts the correct answer, provides an unsupported different answer, fails to provide the answer, or contains only a subset of the information required by the correct answer.

Do not use any external knowledge or subjective inference.

Question: [Question]

Correct Answer: [Correct Answer]

Model Response: [Model Response]

Is the model response correct? Answer yes or no only.

Figure 6: Prompt for LLM judge.

D.3 Downstream LLM Deployment

We evaluate eight downstream LLMs. Six LLMs are used for method training and evaluation: Qwen3.5-9B, Qwen3.5-27B, Qwen3.5-122B-A10B, Gemma-4-26B-A4B-it, GLM-5.2, and DeepSeek-V4-Flash. Two additional LLMs, Llama-3.3-70B-Instruct and GPT-5.1, are used exclusively for unseen-LLM evaluation. LLM responses are obtained through either official provider APIs[1] [2] [3] [4] or a standardized third-party API gateway[5]. We use the same generation setting across all downstream LLMs, with temperature set to 0 to reduce sampling variance. Across all methods, datasets, and downstream LLMs, we use the same prompt template to organize the current query and retrieved conversational history. The template follows the LongMemEval (Wu et al., 2025) input format to ensure a consistent comparison across methods.

[1] https://bailian.console.aliyun.com
[2] https://platform.deepseek.com
[3] https://open.bigmodel.cn
[4] https://platform.openai.com/docs/models/gpt-5.1
[5] https://openrouter.ai

D.4 Baseline Details

For all retrieval-based methods, we use BGE-M3 (Chen et al., 2024) as the retriever. Following prior work (Wu et al., 2025), the turn-level RAG baseline uses historical user queries as retrieval units and retrieves their corresponding complete conversational turns. For RECOMP (Xu et al., 2024), we evaluate its extractive variant, RECOMP-Extr, using the provided `hotpotqa_extractive_compressor` model. For LLMLingua2 (Pan et al., 2024), we use `llmlingua-2-xlm-roberta-large-meetingbank` as its backbone model. All configurations follow the official implementations of the corresponding methods.

E Additional Ablations and Sensitivity to Key Settings

E.1 Discriminator Design Ablation

Table 6: Comparison of unified and separate discriminator designs.

# Disc.	ConvoMem		LongMemEval	
	Acc.↑	Tok.↓	Acc.↑	Tok.↓
1	68.7	624	78.8	1259
2	72.3	643	84.4	1430

Stage 1 partitions historical LLM response candidates into a positive-contribution set and an uncertain set, which have different decision objectives. The retention discriminator determines whether positive candidates remain necessary for the downstream LLM through KEEP/DROP classification, whereas the addition discriminator identifies useful uncertain candidates through ADD/IGNORE classification. Since the two candidate groups differ in both their distributions and decision objectives, we use two structurally identical but independently trained discriminators. For the unified setting, we merge $\mathcal{D}_{\text{keep}}$ and $\mathcal{D}_{\text{add}}$ and train a single discriminator over both candidate groups. As shown in Table 6, this unified design reduces accuracy by 3.6% on CM and 5.6% on LME. This result indicates that a unified decision function is less effective at jointly modeling the retention of necessary positive candidates and the addition of uncertain candidates useful to the downstream LLM.

E.2 Effect of the Semantic Relevance Filtering Threshold τ

Table 7: Sensitivity analysis of the semantic relevance filtering threshold τ. $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ report the proportions of candidates in the three Stage 1 sets. ΔAcc. and ΔTok. are averaged over CM and LME relative to $\tau = -0.05$.

τ	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	ConvoMem		LongMemEval		ΔAcc.	ΔTok.
				Acc.↑	Tok.↓	Acc.↑	Tok.↓		
-0.005	4.0%	10.6%	85.4%	69.0	608	81.9	1244	-2.9	-111
-0.05	4.0%	39.0%	57.0%	72.3	643	84.4	1430	–	–
-0.5	4.0%	75.0%	21.0%	71.1	705	84.1	1867	-0.8	+250

The threshold τ controls the partition between the uncertain set $\mathcal{M}_2$ and the filtered set $\mathcal{M}_3$ in Stage 1. As shown in Table 7, as τ decreases from -0.005 to -0.5, the proportion of candidates assigned to $\mathcal{M}_2$ increases from 10.6% to 75.0%, while that of $\mathcal{M}_3$ decreases from 85.4% to 21.0%. The proportion of positive candidates in $\mathcal{M}_1$ remains unchanged, since τ only controls the partition of candidates with non-positive semantic contribution. A less negative threshold leads to overly aggressive filtering, potentially assigning useful candidates to $\mathcal{M}_3$, whereas a more negative threshold retains many low-contribution candidates in $\mathcal{M}_2$, weakening the filtering role of Stage 1 and increasing the burden on Stage 2 by passing substantially more candidates to it. Accordingly, the average input length across CM and LME increases from 926 to 1,037 and 1,286 tokens as τ changes from -0.005 to -0.05 and -0.5. At $\tau = -0.005$, aggressive filtering reduces average accuracy by 2.9%;

at $\tau = -0.5$, insufficient filtering adds 250 tokens while still reducing accuracy by 0.8%. Overall, $\tau = -0.05$ provides the best balance between retaining potentially useful information and filtering candidates that clearly reduce semantic relevance.

E.3 Performance across Retrieval Depths k

Table 8: Comparison with RAG across different turn-level retrieval depths k on CM and LME.

Method	k	ConvoMem		LongMemEval	
		Acc.↑	Tok.↓	Acc.↑	Tok.↓
RAG	5	60.6	548	73.8	2784
QARS (Ours)	5	64.7 (+4.1)	384 (-164)	75.5 (+1.7)	795 (-1989)
RAG	10	66.9	975	81.0	5340
QARS (Ours)	10	72.3 (+5.4)	643 (-332)	84.4 (+3.4)	1430 (-3910)

As shown in Table 8, our method consistently outperforms RAG across different retrieval depths. At $k = 5$, it improves accuracy by 4.1% on CM and 1.7% on LME while reducing input tokens by 164 and 1,989, respectively. At $k = 10$, the accuracy gains increase to 5.4% and 3.4%, with token reductions of 332 and 3,910. These results show that our method remains effective across retrieval depths, maintaining strong accuracy with substantially fewer input tokens.

E.4 Performance across Different Retrievers

Table 9: Comparison with RAG using different retrievers on CM and LME under turn-level retrieval with $k = 10$.

Method	Retriever	ConvoMem		LongMemEval	
		Acc.↑	Tok.↓	Acc.↑	Tok.↓
RAG	BGE-M3	66.9	975	81.0	5340
QARS (Ours)	BGE-M3	72.3 (+5.4)	643 (-332)	84.4 (+3.4)	1430 (-3910)
RAG	BM25	54.5	1029	67.5	5229
QARS (Ours)	BM25	59.3 (+4.8)	646 (-383)	70.7 (+3.2)	1277 (-3952)
RAG	MPNet	65.7	1038	65.7	5159
QARS (Ours)	MPNet	69.0 (+3.3)	712 (-326)	68.2 (+2.5)	1468 (-3691)

We further evaluate our method with different upstream retrievers, including BGE-M3 (Chen et al., 2024), BM25, and MPNet (`multi-qa-mpnet-base-dot-v1`) (Song et al., 2020), under turn-level retrieval with $k = 10$. As shown in Table 9, our method consistently improves accuracy over the corresponding RAG baseline across all retrievers on both CM and LME, while substantially reducing input tokens. These results demonstrate that our method remains effective across different upstream retrievers.

F Calibration Set Analysis

F.1 Discriminative Capacity of the Diagnostic Calibration Set

To evaluate whether the diagnostic calibration set effectively exposes behavioral differences across downstream LLMs, we compare it with a randomly selected calibration set of the same size. For each set, we report the accuracy of each of the six training LLMs and an Oracle accuracy, where an instance is considered correct if at least one LLM answers it correctly. We report the Oracle Gap, defined as the difference between Oracle accuracy and the best individual-LLM accuracy. To quantify each LLM's contribution to collective recoverability, each Oracle-correct instance contributes one unit of credit, which is divided equally among all LLMs that answer the instance correctly. These credits are aggregated across instances and normalized to obtain each LLM's contribution.

Table 10: Per-LLM statistics for the diagnostic and randomly selected calibration sets. Acc. denotes downstream LLM accuracy, and Contrib. denotes the normalized contribution of each downstream LLM to Oracle-correct instances. All values are percentages.

Downstream LLM	Diagnostic		Random	
	Acc.	Contrib.	Acc.	Contrib.
Qwen3.5-9B	58.2	16.7	66.2	16.3
Qwen3.5-27B	55.5	17.1	66.7	16.5
Qwen3.5-122B	51.6	14.5	66.7	15.3
Gemma-4-26B	59.1	16.9	69.4	17.4
GLM-5.2	62.8	20.0	71.5	18.1
DeepSeek-V4-Flash	50.0	14.8	64.6	16.4
Oracle	94.5	–	86.3	–
Oracle Gap	31.7	–	14.8	–

As shown in Table 10, individual-LLM accuracy is consistently lower on the diagnostic set than on the random set, with differences ranging from 8.0% to 15.1%, whereas Oracle accuracy increases from 86.3% to 94.5%. Consequently, the Oracle Gap increases from 14.8% to 31.7%. This pattern indicates that the diagnostically selected instances are not merely challenging for individual downstream LLMs; rather, different LLMs succeed on different subsets of instances, while the set remains highly recoverable collectively. Moreover, all six LLMs make non-negligible contributions to the diagnostic set, ranging from 14.5% to 20.0%, showing that its collective recoverability is not dominated by any single LLM. These properties induce more differentiated correctness patterns across LLMs, making the diagnostic calibration set well suited for constructing discriminative behavior-based LLM representations. Overall, the results show that diagnostic selection provides more discriminative observations of heterogeneous and complementary context-utilization behaviors than random selection.

F.2 Calibration Set Size and Selection Strategy

Table 11: Impact of calibration set size and selection strategy. M denotes the number of calibration instances.

Selection	M	ConvoMem		LongMemEval	
		Acc.↑	Tok.↓	Acc.↑	Tok.↓
Diagnostic	100	68.7	621	80.9	1271
Diagnostic	200	69.6	635	81.4	1307
Random	438	68.8	614	81.7	1222
Diagnostic	438	72.3	643	84.4	1430

For the size analysis, we progressively select queries following the combined ranking of the two indicators, yielding calibration sets of different sizes M. As shown in Table 11, accuracy on both CM and LME consistently improves as the diagnostic calibration set grows from 100 to 438 instances, suggesting that broader behavioral observations lead to more effective LLM representations. At the same calibration size of $M = 438$, diagnostic selection outperforms random selection by 3.5% and 2.7% on CM and LME, respectively. This improvement indicates that diagnostically selected instances provide more informative behavioral signals for characterizing differences in context utilization across downstream LLMs. Overall, these results show that both calibration set size and instance selection affect representation quality, with diagnostic selection yielding clear advantages over random sampling at the same scale.

F.3 Ablation of Diagnostic Query Selection Indicators

To examine the contribution of the two diagnostic query selection indicators, we construct calibration sets using CtxDis alone, $\mathrm{GainDis}$ alone, or their combination. All other aspects of calibration set construction, including its size, are kept fixed across settings. As shown in Table 12, combining

Table 12: Ablation of the diagnostic query selection indicators on CM and LME. CtxDis and GainDis denote context utilization divergence and answer performance divergence, respectively.

CtxDis	GainDis	ConvoMem		LongMemEval	
		Acc.↑	Tok.↓	Acc.↑	Tok.↓
✓	✗	69.5	627	82.2	1256
✗	✓	69.2	624	81.9	1244
✓	✓	72.3	643	84.4	1430

both indicators achieves the best accuracy on CM and LME. Compared with using CtxDis alone, the combined setting improves accuracy by 2.8% and 2.2% on CM and LME, respectively; compared with using GainDis alone, the improvements are 3.1% and 2.5%. These results indicate that the two indicators provide complementary signals for identifying diagnostically informative queries.

G LLM Representation and Specificity Analysis

G.1 Comparison with ID-Based Representation

Table 13: Comparison between ID-based and behavior-based LLM representations.

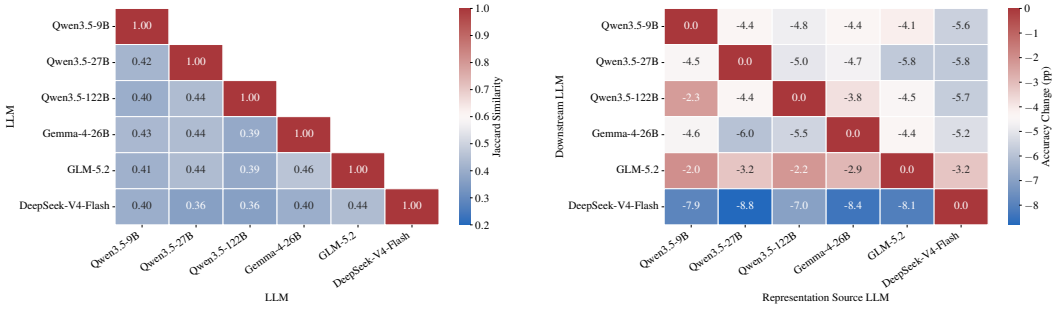
Representation	ConvoMem		LongMemEval	
	Acc.↑	Tok.↓	Acc.↑	Tok.↓
ID-based	70.2	655	82.4	1461
Behavior-based	72.3	643	84.4	1430

To examine the benefit of behavioral calibration for LLM representation, we compare our behavior-based representation with an ID-based alternative. The ID-based representation assigns each LLM a randomly initialized embedding vector, which is jointly optimized with the discriminator training objective. This approach provides a simple identity-level representation without requiring a calibration set or running the LLM on calibration instances. However, since the learned embedding is tied to the LLMs observed during training, it cannot naturally represent unseen LLMs without retraining. As shown in Table 13, the ID-based representation performs worse than our behavior-based representation on both datasets, reducing accuracy by 2.1% on CM and 2.0% on LME. Although the ID-based approach can encode LLM-specific differences observed during training, the performance gap indicates that LLM identity alone is insufficient to capture how different downstream LLMs utilize conversational context. By deriving representations from diagnostic calibration behaviors, our method captures functional differences in context utilization and provides a more informative representation of downstream LLMs, while retaining the ability to represent unseen LLMs without retraining.

G.2 LLM-Specificity Analysis

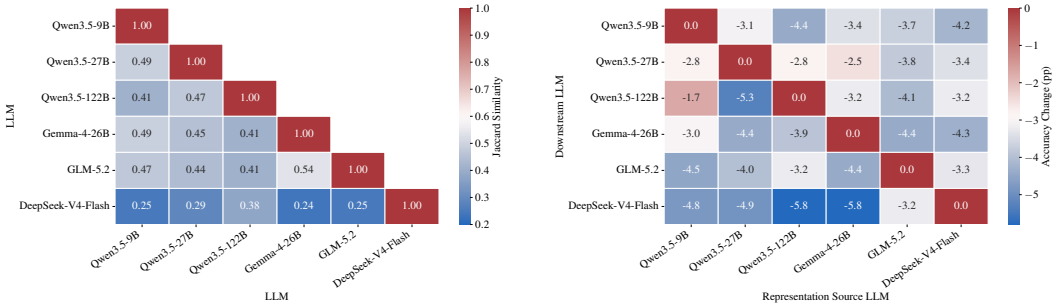
Overlap among Selected LLM Responses. We first examine whether different downstream LLMs rely on the same historical LLM responses. As shown in the left panels of Figures 7 and 8, the selected response sets exhibit relatively low overlap across LLMs. Under the Oracle setting, the average pairwise Jaccard similarity is 0.41, ranging from 0.36 to 0.46. A similar pattern is observed for our method, with an average similarity of 0.40 and a range of 0.24-0.54. Even among the three Qwen3.5 models, the pairwise similarities remain only 0.40-0.44 under Oracle and 0.41-0.49 under our method. These results indicate that different downstream LLMs require different subsets of historical LLM responses, and that our method preserves this LLM-specific selection pattern rather than converging to a universal context.

Cross-LLM Representation Transfer. We further examine whether the behavior representation of one downstream LLM can be directly transferred to another. As shown in the right panel of Figure 7, under the Oracle setting, replacing the target LLM's own representation with that of another LLM consistently decreases answer accuracy, with an average drop of 4.97% and a maximum



(a) Overlap among selected LLM responses. (b) Cross-LLM representation transfer.

Figure 7: LLM-specific context utilization under the Oracle setting.



(a) Overlap among selected LLM responses. (b) Cross-LLM representation transfer.

Figure 8: LLM-specific context utilization under our method.

drop of 8.8%. This result directly demonstrates that different downstream LLMs exhibit distinct context-utilization behaviors and therefore require LLM-specific representations. A similar pattern is observed with our method in Figure 8, where all cross-LLM transfers also lead to accuracy degradation. Together, these results further support the necessity of conditioning historical LLM response selection on the target downstream LLM.

H Efficiency Analysis

Table 14: Online and offline time costs. Online latency is averaged per query after retrieval under turn-level retrieval with $k = 10$ and includes BGE-M3 encoding and context selection. Discriminator training is performed on a single NVIDIA RTX 5090 GPU.

Type	Component	Time Cost
Online	Stage 1	0.33 s per query
	Stage 2	0.12 s per query
	Total selection	0.45 s per query
Offline	Label construction	62.7 min per LLM-dataset pair
	Discriminator training	7.50 min for two discriminators
	LLM calibration	3.86 min per new LLM

We report both online context-selection overhead and offline construction costs. Online efficiency is measured under turn-level retrieval with $k = 10$, with timing starting after retrieval. The measurements therefore exclude retrieval but include BGE-M3 encoding for semantic contribution estimation, semantic relevance filtering, and LLM-conditioned discrimination. Stage 1 requires 0.33 seconds per query on average, while Stage 2 requires 0.12 seconds, resulting in an average total selection overhead of 0.45 seconds per query. The two discriminators jointly require approximately

614 MiB of GPU memory during inference, indicating modest additional computational overhead. These measurements exclude downstream LLM response generation, whose latency depends on the deployed model and API service.

Offline costs include one-time training-label construction and discriminator training, together with behavior calibration for each downstream LLM. We construct training labels through parallel API calls, requiring approximately 62.7 minutes on average per LLM-dataset pair under a single-machine multithreaded setup. Label construction is fully offline and consists of independent API calls; requests within each LLM-dataset pair and different pairs can be processed concurrently, allowing the overall time cost to be further reduced with additional workers or machines and to scale efficiently with increased parallelism. This construction is performed only once before discriminator training and introduces no overhead during online inference. Training both discriminators on a single NVIDIA RTX 5090 GPU takes approximately 7.5 minutes. Constructing an LLM's behavior representation by running it on the diagnostic calibration set requires approximately 3.86 minutes. This lightweight calibration is also performed only once per LLM. For an unseen LLM, the same procedure directly produces its representation without retraining the discriminators and introduces no additional calibration overhead for individual test queries.

I Case Study

We conduct case studies to further examine when historical LLM responses are useful, unnecessary, or potentially interfering for downstream LLMs. As shown in Tables 15–18, we present two representative cases from LME and CM. For each case, the first table shows the retrieved historical memory together with the two-stage response-selection decisions, while the second reports the downstream LLM outputs under the resulting context and comparison setting.

Across the two cases, we observe a clear query-response asymmetry in the retrieved histories shown in Tables 15 and 17. The difference in textual overhead is especially pronounced in Case 1, where historical LLM responses are substantially longer than the corresponding user queries. The cases also reveal an asymmetry in informational role. In Case 1, user queries introduce user-specific goals, preferences, and needs, whereas LLM responses elaborate on these requests using the LLM's existing knowledge and supply factual background, including brand information relevant to the current query. In Case 2, user queries directly introduce new information about evolving external states, while the corresponding LLM responses largely confirm, restate, or reformulate these updates without adding information necessary for the current query. Together, these cases motivate retaining historical user queries as memory anchors and selectively incorporating historical LLM responses according to their utility for the current query and downstream LLM.

Case 1 illustrates how the utility of historical LLM responses varies across downstream LLMs under final-answer correctness evaluation. Our method retains fewer responses for GLM-5.2 than for Qwen3.5-9B. GLM-5.2 correctly identifies Veja without the relevant historical response, drawing on its existing knowledge as indicated in its output. Qwen3.5-9B abstains under the Response-Free setting but answers correctly with the context selected by QARS, which includes the response naming Veja. Thus, explicit historical response evidence helps Qwen3.5-9B produce a correct answer, while GLM-5.2 can answer correctly without it in this example. These outcomes motivate selecting historical responses according to their utility for the target downstream LLM. The selection pattern also illustrates the two-stage process: Stage 1 narrows the candidate set through semantic relevance filtering, while Stage 2 makes LLM-specific selection decisions over the remaining candidates.

In Case 2, the historical user queries already contain the evidence needed for the answer, while some LLM responses add information that is not necessary for the answer and may interfere with downstream generation. RAG retains these responses and answers incorrectly, whereas our method removes the interfering responses and answers correctly. Answering requires integrating multiple updates over time rather than relying on an earlier intermediate response, showing that some historical LLM responses may be unnecessary or harmful and motivating selective retention.

Table 15: Case 1 from LME. Boxes show Stage 1 decisions (**Positive**/ **Uncertain**/ **Filtered**) → Stage 2 decisions (**Keep**/ **Drop**/ **Add**/ **Ignore**/ **Filtered**). Some nonessential details are omitted.

Case 1: Retrieved Historical Memory and QARS Selection Decisions

RETRIEVED UNIT 1
User query: I've been meaning to get my old Adidas leggings replaced, do you have any recommendations for good athletic wear brands?
LLM response: Upgrading your athletic wear game, eh? There are many great brands out there, depending on your specific needs, preferences, and activities. For running and high-intensity workouts, options include Nike, Under Armour, and New Balance; for yoga and low-impact activities, Lululemon, Athleta, and Prana are popular choices. Outdoor and sustainable options include Patagonia, Outdoor Voices, Girlfriend Collective, and People Tree. *[Additional categories and recommendations omitted.]*
GLM-5.2: Uncertain → Ignore | Qwen3.5-9B: Uncertain → Add

RETRIEVED UNIT 2
User query: I'm also thinking of getting a new pair of Converse sneakers to replace my old worn-out ones. Do you have any recommendations for good shoes for casual wear?
LLM response: New kicks, eh? Converse is a great choice, but there are many other options for casual wear. Classic choices include Converse Chuck Taylor All Star, Vans Old Skool, and Adidas Superstar, while comfort-focused options include New Balance 574, Clarks Cloud Steppers, and TOMS Deck Shoes. Fashion-forward alternatives include Gucci Ace, Prada Sport, and Saint Laurent Court Classic sneakers. *[Additional recommendations and purchasing advice omitted.]*
GLM-5.2: Filtered → Filtered | Qwen3.5-9B: Filtered → Filtered

RETRIEVED UNIT 3
User query: No specific preferences yet, I'm still open to exploring different options. But I do know I want something that's reliable and accurate, since I'll be wearing it daily. I'll definitely consider the brands you mentioned and do some research to find the best fit for me. Thanks for the recommendations!
LLM response: You're welcome! Reliability and accuracy are essential features to consider when choosing a watch, especially if you plan to wear it daily. You should also consider your personal style, lifestyle, and the occasions where the watch will be worn, and compare reviews and features before making a decision. *[Closing advice omitted.]*
GLM-5.2: Filtered → Filtered | Qwen3.5-9B: Filtered → Filtered

RETRIEVED UNIT 4
User query: I think I'll check out Amazon and Zazzle for some options. I like the idea of customizing my own mug. Do you have any suggestions on how to choose the right material for my new mug?
LLM response: Choosing the right material depends on personal preferences, needs, and lifestyle. Ceramic offers durability and heat retention; stoneware has a more rustic feel; glass is non-porous and easy to clean; porcelain combines strength and a refined appearance; silicone is flexible and travel-friendly; and insulated mugs retain temperature for longer. Important factors include durability, ease of cleaning, heat retention, aesthetics, and lifestyle. *[Additional material details omitted.]*
GLM-5.2: Filtered → Filtered | Qwen3.5-9B: Filtered → Filtered

RETRIEVED UNIT 5
User query: I'm also considering taking up the ukulele, do you have any recommendations for a beginner-friendly ukulele brand and model?
LLM response: The ukulele is an excellent instrument for beginners. Recommended options include the Kala KA-15S Mahogany Soprano Ukulele, Fender Seaside Ukulele, Lanikai LU-21T Tiki Concert Ukulele, and Cordoba 20TM Tenor Ukulele. When choosing a model, important factors include size, material, action, and tuning pegs. *[Detailed descriptions, prices, and additional advice omitted.]*
GLM-5.2: Filtered → Filtered | Qwen3.5-9B: Filtered → Filtered

RETRIEVED UNIT 6
User query: That makes sense. I've always wondered why high-end fashion brands are so expensive.
LLM response: High-end fashion brands can be more expensive because they often use premium materials such as silk, cashmere, and high-grade leather, employ skilled artisans and designers, produce more exclusive items in limited quantities, and invest substantially in marketing and branding. *[Further explanation omitted.]*
GLM-5.2: Uncertain → Ignore | Qwen3.5-9B: Uncertain → Ignore

RETRIEVED UNIT 7
User query: I agree, it's important to consider my budget before investing in high-end brands. But I do love the idea of owning a piece that lasts a long time and has a unique design. Do you have any recommendations for high-end brands that prioritize sustainability?
LLM response: Sure! Here are a few high-end fashion brands known for their commitment to sustainability:
1. Stella McCartney – uses environmentally friendly materials such as organic cotton and recycled polyester.
2. Eileen Fisher – uses organic cotton, hemp, and linen and operates a take-back program.
3. Patagonia – uses eco-friendly fabrics and invests in conservation and environmental protection.
4. Reformation – uses eco-friendly materials and promotes supply-chain transparency.
5. Veja – produces eco-friendly sneakers using organic cotton, recycled plastic bottles, and wild rubber sourced from the Amazon rainforest.

[Additional closing remarks omitted.]
GLM-5.2: Uncertain → Ignore | Qwen3.5-9B: Uncertain → Add

RETRIEVED UNIT 8
User query: I really like the idea of buying from brands that are committed to sustainability. I'll definitely check those out. Do you know if any of these brands have sales or discounts on their products?
LLM response: High-end fashion brands may offer seasonal sales or promotions, although these can be limited. Other options include shopping during off-season periods, purchasing gently used items from marketplaces such as The RealReal or Vestiaire Collective, and following brand newsletters or social media for promotions. *[Additional advice omitted.]*
GLM-5.2: Uncertain → Ignore | Qwen3.5-9B: Uncertain → Ignore

RETRIEVED UNIT 9
User query: I see, I never realized how much goes into high-end fashion pieces. Do you think investing in high-end brands is worth it?
LLM response: Investing in high-end fashion may be worthwhile if you value high-quality materials, unique designs, craftsmanship, sustainability, durability, and longevity. However, the decision depends on personal preferences, values, and budget. *[Additional discussion omitted.]*
GLM-5.2: Uncertain → Ignore | Qwen3.5-9B: Uncertain → Add

RETRIEVED UNIT 10
User query: How does a high-end fashion brand differ from fast fashion?
LLM response: High-end fashion and fast fashion differ in several respects: high-end brands generally prioritize quality, craftsmanship, distinctive design, smaller-scale production, and higher prices. They may also place greater emphasis on sustainable materials and ethical manufacturing, whereas fast-fashion brands typically focus on large-scale, rapid production. *[Additional explanation omitted.]*
GLM-5.2: Uncertain → Ignore | Qwen3.5-9B: Uncertain → Ignore

Table 16: Case 1 from LME. Parentheses indicate the context setting (QARS/Response-Free).

Case 1: Downstream LLM Outputs

Current query: I was going through our previous conversation about high-end fashion brands, and I was wondering if you could remind me of the brand that uses wild rubber sourced from the Amazon rainforest?
Gold answer: Veja

GLM-5.2 (QARS) — Correct

Based on the provided chat history, I cannot remind you of the specific brand, as the assistant's responses from our previous conversation were not included in the text. However, the brand you are likely thinking of is **Veja**, which is well-known for using wild rubber sourced from the Amazon rainforest.

Qwen3.5-9B (QARS) — Correct

I can directly link the specific material description to the brand name found in Session 7.

- The query asks for the brand using "wild rubber sourced from the Amazon rainforest."
- In the assistant's response within Session 7, item number 5 explicitly states: "Veja – This French brand produces eco-friendly sneakers using organic cotton, recycled plastic bottles, and **wild rubber sourced from the Amazon rainforest**."
- No other brand in the list or the rest of the conversation history contains this specific detail.

The brand that uses wild rubber sourced from the Amazon rainforest is **Veja**.

Qwen3.5-9B (Response-Free) — Incorrect

I cannot answer this question based on the provided history chats. While the conversation history (Sessions 6–10) shows that you asked for recommendations for high-end fashion brands that prioritize sustainability, the **assistant's responses** containing the specific brand names and details (such as the one using wild rubber from the Amazon rainforest) are not included in the text provided.

Table 17: Case 2 from CM. Boxes show Stage 1 decisions (**Positive**/ **Uncertain**/ **Filtered**) → Stage 2 decisions (**Keep**/ **Drop**/ **Add**/ **Ignore**/ **Filtered**).

Case 2: Retrieved Historical Memory and QARS Selection Decisions

RETRIEVED UNIT 1
User query: Yes, please. It's for Greenleaf Organics. The kickoff meeting this morning went really well. I'm feeling optimistic about this one.
LLM response: That's great to hear. I've updated the calendar event notes to specify 'Greenleaf Organics Onboarding'. What's the first task on your list for them?
Gemma-4-26B: **Uncertain** → **Ignore**

RETRIEVED UNIT 2
User query: Let's call it 'Greenleaf Organics - Q3 2024 Onboarding'. Pretty standard.
LLM response: Project 'Greenleaf Organics - Q3 2024 Onboarding' has been created. I've populated it with the standard onboarding task list. Would you like to review or modify the tasks?
Gemma-4-26B: **Uncertain** → **Ignore**

RETRIEVED UNIT 3
User query: Yes, a crucial one. I need to add the primary contact information. Just wrapped up the onboarding kickoff for Greenleaf Organics. Their main point of contact and designated power user is Mark.
LLM response: Thank you for that information. I am searching the CRM for a contact named Mark at Greenleaf Organics. Do you have his last name?
Gemma-4-26B: **Uncertain** → **Ignore**

RETRIEVED UNIT 4
User query: Good morning. Can you pull up the active case list for Greenleaf Organics? I just got a slightly confusing email and want to check if a new ticket was created.
LLM response: Good morning, Emma. I am pulling up the case list for Greenleaf Organics. There is one new active case, created 35 minutes ago. The subject is 'Data Sync Error - Urgent'.
Gemma-4-26B: **Uncertain** → **Ignore**

RETRIEVED UNIT 5
User query: Hold on, let me re-read her email signature... ah, that explains it. Quick update for Greenleaf Organics: there's been a personnel change. Susan will be taking over all power user responsibilities from Mark, effective today.
LLM response: Thank you for the clarification. This is an important update. Shall I update the CRM to reflect this change? I will need to set Susan Wright as the new Power User and Primary Contact.
Gemma-4-26B: **Uncertain** → **Ignore**

RETRIEVED UNIT 6
User query: Great. I want to schedule a final check-in call with each of them. Let's start with Greenleaf Organics. Please pull up their main contact info.
LLM response: The primary contact and power user for Greenleaf Organics is Susan Wright.
Gemma-4-26B: **Positive** → **Drop**

RETRIEVED UNIT 7
User query: Okay, I see what's happening. Got a notification regarding the Greenleaf account. The current power user is going on extended leave, so Tom will be the interim contact during this period.
LLM response: Thank you for the update. This changes the plan. I will need to update the contact status in the CRM. Do you have a last name for Tom?
Gemma-4-26B: **Uncertain** → **Ignore**

RETRIEVED UNIT 8
User query: Morning. Can you pull up the Greenleaf Organics account file? I need to review the status of their power user.
LLM response: Good morning, Emma. Here is the file for Greenleaf Organics. The current contact is listed as Tom Chen, designated as 'Interim Primary Contact'. Susan Wright's status is 'On Leave'.
Gemma-4-26B: **Filtered** → **Filtered**

RETRIEVED UNIT 9
User query: It certainly does. Final update for Greenleaf Organics: they've hired Chloe as the new permanent replacement for the role Tom was covering.
LLM response: This is great news. It provides a clear path forward. I assume Chloe is Chloe Davis, who is mentioned in the email body?
Gemma-4-26B: **Uncertain** → **Ignore**

RETRIEVED UNIT 10
User query: Yes, do that. That's the best way to handle it. Title it 'Welcome & Account Handoff: Greenleaf Organics'.
LLM response: The invitation has been drafted and sent to you for final approval before it goes to Chloe and David.
Gemma-4-26B: **Uncertain** → **Ignore**

Table 18: Case 2 from CM. Parentheses indicate the context setting (QARS/RAG).

Case 2: Downstream LLM Outputs

Current query: For the Greenleaf Organics account, who is the designated power user?
Gold answer: Chloe is the designated power user for Greenleaf Organics.

Gemma-4-26B (QARS) — Correct

1. Initially, the power user was **Mark**.
2. The responsibility was then transferred from Mark to **Susan**.
3. During an extended leave, **Tom** took over as the interim contact.
4. Finally, **Chloe** was hired as the permanent replacement for the role Tom was covering.

Answer: The current designated power user for Greenleaf Organics is **Chloe**.

Gemma-4-26B (RAG) — Incorrect

To determine the current designated power user, we must follow the chronological personnel changes described in the sessions:

1. Initial State: Mark was the designated power user.
2. First Change: Susan Wright was set to take over the power user responsibilities from Mark.
3. Interim Change: During a leave of absence, Tom Chen became the interim contact/power user.
4. Final State: Session 6 identifies Susan Wright as the primary contact and power user. Session 9 states that Chloe was hired to replace Tom, which is interpreted as ending the interim period while leaving Susan Wright as the designated power user.

Answer: The designated power user for Greenleaf Organics is **Susan Wright**.