

Mem-ABC: Attitude-Aware Agent Memory for Personalized Question Answering

Yinghan Shi¹, Kai Fu¹, Quanyou Chu¹, Xinyu Liu¹, Ming Du¹, Hongya Wang¹,
Xiaojun Meng², Jiansheng Wei², Yanghua Xiao³, Bo Xu^{1*}

¹School of Information and Intelligent Science, Donghua University

²Huawei Large Model Data Technology Lab ³Shanghai Key Laboratory of Data Science,
College of Computer Science and Artificial Intelligence, Fudan University
{2242829, 2242825, 2242865, 2242850}@mail.dhu.edu.cn
{duming, hywang}@dhu.edu.cn, {xiaojun.meng, weijiansheng}@huawei.com
shawyh@fudan.edu.cn, xubo@dhu.edu.cn

Abstract

Personalized Question Answering (PQA) requires models to shift from a fact-centric to a user-centric paradigm, inferring implicit intents from historical interactions. However, standard Retrieval-Augmented Generation (RAG) struggles with narrative noise and fragmented semantics. Similarly, while Agent Memory was proposed to maintain long-term consistency, existing implementations predominantly focus on accumulating objective facts, neglecting the subjective attitudes essential for personalization. We argue that for PQA, while objective facts provide the necessary foundational context, relying solely on them is insufficient; they must be augmented by explicit modeling of subjective attitudes to capture a comprehensive user profile. To address this, we introduce **Mem-ABC**, an Attitude-Aware Agent Memory framework grounded in the Tripartite Model of Attitudes (Affect, Behavior, and Cognition). Mem-ABC employs a two-stage offline process: first autonomously refining raw history into general memories to eliminate noise, and subsequently reorganizing them into a structured attitude-aware memory. This structure enables the agent to explicitly retrieve context aligned with user psychology. Experiments on the LaMP-QA benchmark demonstrate that Mem-ABC consistently outperforms strong baselines across multiple domains.

1 Introduction

Recently, personalized question answering (PQA) has emerged as a key research direction (Salemi and Zamani, 2025a), shifting intelligent information systems from a fact-centric to a user-centric paradigm. Unlike traditional question answering (QA) systems that seek standardized factual answers, the core challenge of PQA is that the cor-

rectness of a response is not absolute but highly dependent on the user’s background, preferences, and past experiences. For instance, when asking *Where should I go for the weekend?*, an adventure enthusiast might expect recommendations for hiking trails and camping sites, whereas a user who values peace and comfort would prioritize quiet resorts or museums. This difference requires the model not only to understand explicit query semantics but also to infer implicit user intents and preferences from accumulated user history.

Existing solutions predominantly adopt the Retrieval Augmented Generation (RAG) paradigm (Lewis et al., 2020), which retrieves the top- k most similar records from user history as personalized context. However, as illustrated in Figure 1(a), relying on raw user history imposes two key limitations on complex personalized reasoning: 1) Distraction from Irrelevant Details: Raw history contains narrative noise irrelevant to the current query, which can waste the model’s processing capacity and mislead it from the user’s real intent. 2) Difficulty in Integrating Clues: User preferences are often implicit and dispersed throughout the history, rather than concentrated in a single entry. Gathering these scattered clues requires complex multi-step reasoning, which remains a weak point for current LLMs.

Recent advancements in structured agent memory (Chhikara et al., 2025; Xu et al., 2025; Rasmussen et al., 2025; Zhang et al., 2025; Tan et al., 2025), proposed for tasks like long-text understanding and multi-turn dialogue, offer new possibilities for addressing these issues. By autonomously refining static history into structured storage, these methods enable complex reasoning and maintain cognitive consistency across long-term interactions. However, existing implementations predominantly focus on extracting objective facts via predefined schemas, as illustrated in Figure 1(b). While these facts provide a foundational context for user inter-

*Corresponding author. The work is supported by the Donghua University 2026 Cultivation Project of Discipline Innovation xkcx-202622.

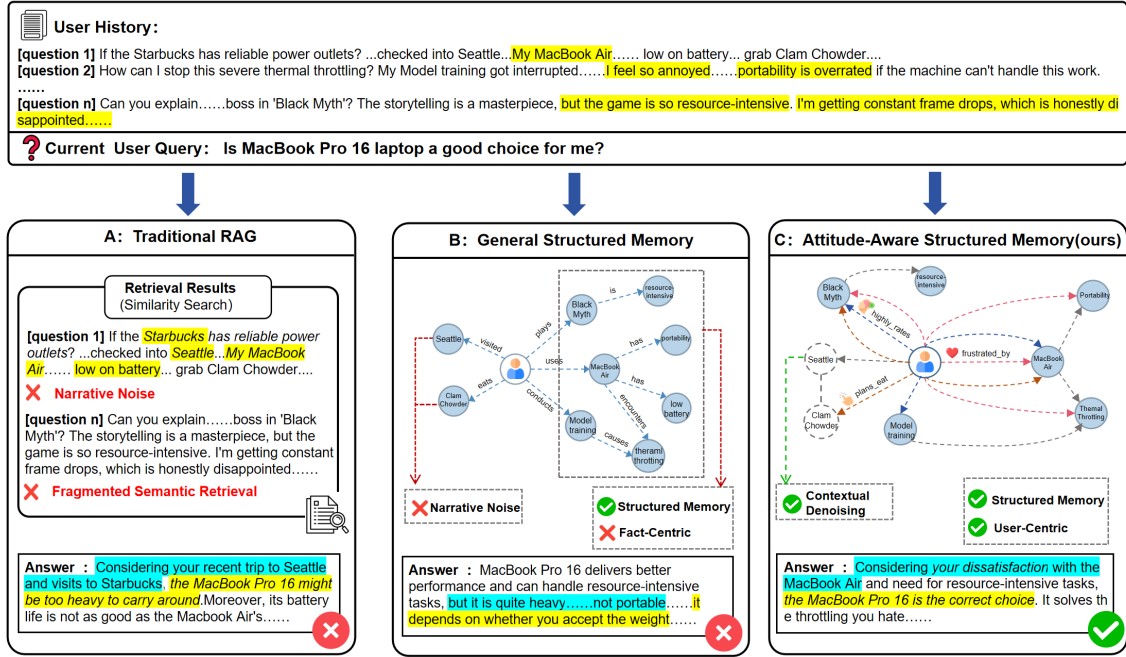


Figure 1: **Comparison of three paradigms.** (A) Traditional RAG suffers from susceptibility to narrative noise and fragmented retrieval. (B) General Structured Memory relies on indiscriminate fact accumulation, remaining agnostic to user attitude. (C) Mem-ABC achieves robust contextual denoising and explicit attitude modeling, enabling precise user-centric reasoning.

actions, this purely fact-centric paradigm is fundamentally insufficient for deep personalization. **We argue that for PQA tasks, objective fact accumulation must be augmented by the explicit modeling of subjective attitudes, as these attitudes serve as the critical psychological determinants for inferring latent user intent.**

To address these challenges, we propose **Mem-ABC**, a novel Attitude-Aware Agent Memory framework. Our approach involves an initial distillation of raw user history into refined general memories to minimize narrative noise. Subsequently, we ground our memory construction in the *Tripartite Model of Attitudes* (ABC Model) from social psychology (Rosenberg, 1960; Breckler, 1984; Zanna and Rempel, 1988). **This theory posits that a complete human attitude is not single-dimensional but a composite construct formed by the interplay of three components: *Affect, Behavior, and Cognition*.** Guided by this theoretical foundation, we reorganize discrete memory fragments into a cohesive structure that explicitly captures these three dimensions. This framework empowers the agent to construct a holistic user profile and infer implicit intents, significantly improving user satisfaction.

Specifically, our contributions are as follows:

- To the best of our knowledge, we are the first to

incorporate the psychological *Tripartite Model of Attitudes* into the Personalized Question Answering task. This perspective shifts the focus of agent memory from objective fact accumulation to subjective attitude modeling, enabling a deeper understanding of user intent.

- We propose **Mem-ABC**, a novel agent memory framework that employs a sequential pipeline to filter narrative noise via General Memory Extraction and construct structured, user-centric memory via Attitude-Aware Memory Extraction.
- Extensive experiments on the LaMP-QA benchmark demonstrate that our method outperforms strong baseline models. Furthermore, we conduct comprehensive ablation studies and case studies to validate the effectiveness and necessity of each component within our framework.

2 Related Work

2.1 Personalized Question Answering

Personalized Question Answering (PQA) aims to align responses with the user’s unique background and preferences (Maragheh et al., 2025; Kumar et al., 2024; Salemi et al., 2024; Shi et al., 2025). Methodologically, approaches range from model fine-tuning to retrieval-based strategies. For instance, the VAC framework (Salemi and Zamani,

2025b) employs a reinforcement learning paradigm, utilizing natural language feedback to fine-tune the policy model for more nuanced personalized generation.

However, the prevailing paradigm remains Retrieval-Augmented Generation (RAG), which enhances generation by retrieving relevant context from user history. To overcome the limitations of direct retrieval, POT (Salemi et al., 2025) introduces an inference-time mechanism that allows the model to explore multiple reasoning paths. Despite these advancements, existing works largely treat user history as a static resource. This paper focuses on addressing this gap by enabling agents to actively manage personalized context.

2.2 Agent Memory Systems

Agent memory systems empower LLMs to maintain long-term consistency by autonomously managing information. Research in this domain generally adopts either an unstructured paradigm, optimizing text segment operations via Reinforcement Learning (Yu et al., 2025; Yan et al., 2025; Wang et al., 2025; Tan et al., 2025), or a structured paradigm, utilizing graphs to capture complex dependencies (Rasmussen et al., 2025; Chhikara et al., 2025). While effective for logical reasoning, these systems remain predominantly fact-centric. They often overlook subjective attitudes—a critical dimension for deep personalization that this paper seeks to model.

2.3 The Tripartite Model of Attitudes

Originating from social psychology, the Tripartite Model of Attitudes (ABC Model) posits that attitudes comprise three interrelated components: **Affect, Behavior, and Cognition** (Rosenberg, 1960; Breckler, 1984). Modern interpretations further define attitude as a dynamic evaluation tendency derived from these components (Zanna and Rempel, 1988). We leverage this theoretical foundation to structure agent memory, enabling the inference of latent user profiles from fragmented interaction data.

3 Overview

3.1 Problem Formulation

We formally define the personalized question answering task as follows: Given a user u and a current query x_u , the user’s history set is denoted as $H_u = \{h_i\}_{i=1}^{n_u}$, where n_u represents the total

number of historical interactions for user u . Each history h_i is represented as a tuple $h_i = (q_i, r_i)$, comprising a question title q_i and a detailed question body (question narrative) r_i . The objective is to utilize a question answering model LLM to generate a personalized response $\hat{y}_u = LLM(x_u, H_u)$ for the current query x_u by leveraging information from the user history. For more comprehensive evaluation details, please refer to Appendix A.2.

3.2 Framework

The overall architecture of our Mem-ABC framework is illustrated in Figure 2. The entire process consists of two core phases: **Offline Memory Construction** and **Online Personalized Question Answering**.

In the offline phase, we focus on transforming unstructured user history into a structured attitude-aware agent memory. This phase proceeds through two sequential stages: General Memory Extraction and Attitude-Aware Memory Extraction. In Stage 1, the agent first extracts Candidate Memories from the raw user history and a Dynamic Memory Decision Tool filters these candidates against existing memories, performing incremental updates to construct a refined set of General Memories.

In Stage 2, taking the General Memories as input, we first execute Dual-Memory Entity Extraction to identify and consolidate entities. Subsequently, we perform attitude-aware relation extraction by defining composite relation types that guide the instantiation of entity connections, thereby synthesizing discrete information into high-density Attitude-Aware Memories.

In the online phase, upon receiving a current user query, the system employs memory hybrid retrieval over the constructed Attitude-Aware Memories to retrieve relevant context, directly guiding the LLM to generate a personalized response.

4 Methodology

In this section, we formally present the detailed implementation of the Mem-ABC framework.

4.1 General Memory Extraction

As the first stage of the Offline Memory Construction phase, the primary objective of this stage is to distill raw user history into a noise-free General Memories set $\mathcal{M}_g$. Inspired by the incremental extraction paradigm of Mem0 (Chhikara et al., 2025), we adopt a similar approach to refine unstructured

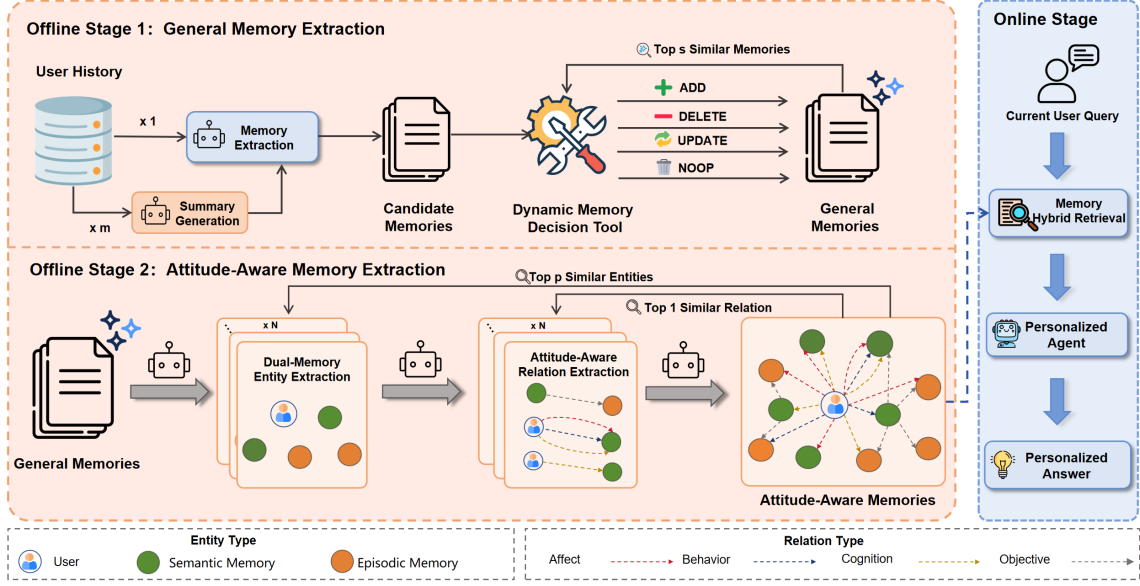


Figure 2: **The overall architecture of the Mem-ABC framework.** Our approach transforms raw user history into structured memory through a **two-stage offline pipeline**: (1) **General Memory Extraction** distills text to mitigate narrative noise; and (2) **Attitude-Aware Memory Extraction** constructs a graph grounded in the *Tripartite Attitude Model*. This structured memory explicitly models user subjectivity to guide personalized generation.

text. However, distinct from Mem0 which treats this refined text as the final storage, we designate $\mathcal{M}_g$ as a foundational intermediate layer, serving as the clean contextual basis for the subsequent attitude-aware modeling in Section 4.2.

4.1.1 Candidate General Memory Extraction

In this section, we utilize global summary generation to assist the extraction agent $\mathcal{E}_g$ in extracting candidate general memories. Specifically, for each history $h_i \in H_u$, $\mathcal{E}_g$ identifies and extracts a set of candidate memories $F_i = \{f_1^{(i)}, \dots, f_m^{(i)}\}$.

To ensure $\mathcal{E}_g$ captures cross-document long-term dependencies and accurately identifies the central topic of the current document, we concurrently maintain a global summary S_g . It is maintained through an incremental update strategy, triggered every $m_s = 5$ new history records. Importantly, the extraction process is designed to capture core, long-term information while filtering out temporary conversational noise. Guided by prompt instructions, $\mathcal{E}_g$ focuses on specific user preferences, stated goals, and personal experiences, strictly ignoring generic chitchat or redundant information already covered by S_g . The extraction process is formally defined as:

$$F_i = \mathcal{E}_g(h_i \mid S_g) \quad (1)$$

4.1.2 Dynamic Memory Decision Tool

To maintain the timeliness and logical consistency of $\mathcal{M}_g$, the dynamic memory decision tool is tasked with performing strict consistency verification and deduplication for each newly extracted candidate memory $f_j^{(i)} \in F_i$.

Specifically, the candidate entry $f_j^{(i)}$ is processed through a function-calling interface. The tool first retrieves the top- s most relevant existing entries $\mathcal{M}_{top-s}$ from $\mathcal{M}_g$ as context. Based on this retrieval, it empowers the agent to autonomously reason over the current history h_i and decide on a precise operation $\alpha \in \{\text{Add}, \text{Update}, \text{Delete}, \text{Noop}\}$ to evolve the memory state. This decision process is formalized as:

$$\alpha = \pi(f_j^{(i)}, \mathcal{M}_{top-s} \mid h_i, S_g), \quad \alpha \in \Omega \quad (2)$$

where the operation set Ω is defined as:

- **ADD**: When no semantically equivalent entry exists in $\mathcal{M}_{top-s}$, instantiate $f_j^{(i)}$ as a new memory entry.
- **UPDATE**: When $f_j^{(i)}$ provides critical supplementary details or state changes for an existing entry, merge the new and old information.
- **DELETE**: Because user history is processed chronologically, when $f_j^{(i)}$ explicitly contradicts an existing entry, the agent removes the obsolete entry. This incremental update mechanism en-

sures that evolving preferences safely supersede outdated ones.

- NOOP: When $f_j^{(i)}$ contains redundant or low-value information that already exists, discard the candidate and make no changes to $\mathcal{M}_g$.

4.2 Attitude-Aware Memory Extraction

As the second stage of the Offline Memory Construction phase, the primary objective of this stage is to transform the refined General Memories $\mathcal{M}_g$ into a structured Attitude-Aware Agent Memory $\mathcal{G}_u$. We formalize this memory as a graph structure $\mathcal{G}_u = (\mathcal{V}, \mathcal{R}, \Phi)$, where $\mathcal{V}$ denotes the set of entities, $\mathcal{R}$ denotes the set of relations, and Φ represents the composite relation types.

4.2.1 Dual-Memory Entity Extraction

Grounded in the cognitive consensus that human memory comprises two fundamental components—*episodic memory* and *semantic memory*—this module operationalizes this dual-memory paradigm for entity extraction. To align with this theoretical foundation while maintaining a user-centric structure, we categorize extracted entities into three primary classes:

- USER ENTITY (v_u): The central identity representing the user.
- SEMANTIC ENTITIES (v_s): Entities representing long-term, stable conceptual knowledge.
- EPISODIC ENTITIES (v_e): Entities representing specific event fragments or temporal experiences.

The complete set of entities within the memory is formally defined as:

$$\mathcal{V} = v_u \cup v_s \cup v_e \quad (3)$$

To ensure precision and maintain a clean memory state, we implement a stratified *Entity Resolution* strategy based on the similarity score s between a candidate’s name embedding and existing entities. Specifically, we first retrieve the top- k similar entities from the memory using vector embeddings. Subsequently, we define a merge threshold τ_m and a disambiguation threshold τ_a to execute a tiered processing logic: (1) Hard Merge: When $s \geq \tau_m$, the system identifies the candidate as a redundant mention of an existing entity and directly consolidates them into a single unique entity; (2) Soft Merge: When $s \in [\tau_a, \tau_m)$, the candidate, the existing entity, and the current context are fed into the LLM for semantic disambiguation to determine whether they should be unified; and (3) Create:

When $s < \tau_a$, the candidate is instantiated as a new entity within $\mathcal{G}_u$.

This tiered approach effectively reduces redundant LLM inference overhead while ensuring the structural integrity and accuracy of the memory.

4.2.2 Attitude-Aware Relation Extraction

This component functions as the core of our framework by operationalizing the abstract *ABC Model* into a concrete schema for Attitude-Aware Agent Memory construction. This theory posits that a user’s attitude toward an entity is not single-dimensional but comprises three interrelated components: *Affect* (emotional responses), *Behavior* (actions or intentions), and *Cognition* (beliefs or value judgments).

To model this complexity, we define the composite relation types Φ as the union of a subjective attitude set Φ_{ABC} and an objective factual set Φ_{obj} :

$$\begin{aligned} \Phi &= \Phi_{ABC} \cup \Phi_{obj}, \\ \text{where } \Phi_{ABC} &= \{\phi_a, \phi_b, \phi_c\} \end{aligned} \quad (4)$$

In this definition, ϕ_a , ϕ_b , and ϕ_c correspond to the specific relation categories of affect, behavior, and cognition, respectively.

The relation extraction agent $\mathcal{E}_r$ transforms general memories $f \in \mathcal{M}_g$ into structured relations. For user-centric interactions, $\mathcal{E}_r$ is guided by prompts to sample and instantiate corresponding relations from the attitude set Φ_{ABC} between the user entity v_u and an external entity v_j . This generation process is formally expressed as:

$$r_{u,j} = \mathcal{E}_r(v_u, v_j \mid f, \Phi_{ABC}) \quad (5)$$

The generated relation $r_{u,j}$ carries a single, explicit attribute where $\text{type}(r_{u,j}) \in \Phi_{ABC}$.

Our framework supports a multi-relation mechanism, allowing the same entity pair (v_u, v_j) to possess multiple concurrent attitude types. This effectively handles complex attitude conflicts (e.g., user cognitively values a product but affectively dislikes its flavor), passing these non-mutually exclusive dimensions as structured context for the LLM to synthesize a balanced response. Conversely, for relations between non-user entities $(v_i, v_j \neq v_u)$, $\mathcal{E}_r$ restricts the extraction to the objective set Φ_{obj} to maintain factual accuracy:

$$r_{i,j} = \mathcal{E}_r(v_i, v_j \mid f, \Phi_{obj}) \quad (6)$$

To prevent redundancy and avoid costly full-graph rebuilds, the structured memory is updated

incrementally via database merge operations. New relations are instantiated—or overwrite obsolete ones—only if their semantic similarity s to existing edges satisfies $s < \tau_p$, ensuring the graph expands efficiently with novel evaluations. To ensure the reliability of the extracted relations, we conducted a supplementary human evaluation, confirming that our framework captures grounded user attitudes rather than hallucinations (detailed setups and quantitative results are provided in Appendix D).

4.3 Personalized Question Answering

During inference, we retrieve relevant context for query x_u using a dual-route vector retrieval strategy. Specifically, we simultaneously perform similarity retrieval across both the entity and relation embedding spaces. The candidate triples associated with these retrieved elements are then unioned, ranked by their cosine similarity scores, and strictly truncated to the top- t results before being linearized into a coherent textual representation. These statements form a high-density personalized context C , effectively aligning the structured graph data with the textual input requirements of the LLM.

Finally, conditioned on the query x_u and the linearized context C , the LLM θ generates the optimal response $\hat{y}_u$ by maximizing the joint probability of the output token sequence $\mathbf{y} = \{y_1, \dots, y_L\}$:

$$\begin{aligned} \hat{y}_u &= \arg \max_{\mathbf{y}} P_{\theta}(\mathbf{y} \mid x_u, C) \\ &= \arg \max_{\mathbf{y}} \prod_{t=1}^L P_{\theta}(y_t \mid y_{<t}, x_u, C) \end{aligned} \quad (7)$$

where y_t denotes the t -th token in the response sequence, $y_{<t}$ represents the preceding context tokens, and L is the length of the generated response.

5 Experiments

5.1 Datasets

We conducted experiments on the LaMP-QA benchmark (Salemi and Zamani, 2025a), which is currently the only publicly available personalized question-answering dataset. The LaMP-QA benchmark covers three distinct domains: (1) Art and Entertainment, (2) Lifestyle and Personal Development, and (3) Society and Culture. For more comprehensive details, please refer to Appendix A.1.

5.2 Metrics

To ensure a fair comparison with previous works, we follow the evaluation settings described in LaMP-QA (Salemi and Zamani, 2025a) by adopting an aspect-based evaluation framework. Specifically, we employ Qwen-2.5-32B-Instruct¹ as the primary evaluator to assess the extent to which the generated responses cover the extracted personalized aspects. To mitigate evaluator bias and improve evaluation reliability, we conduct cross-model verification with Llama-3.1-70B and human evaluation. Detailed scoring rubrics, evaluation protocols, and human preference results are reported in Appendices A and B.

5.3 Baselines

We evaluate Mem-ABC against five representative baselines:

- (1) **No-Personalized:** Directly prompts the LLM using its internal parametric knowledge, without any user history (Salemi and Zamani, 2025a).
- (2) **RAG-Personalized:** The standard Retrieval-Augmented Generation approach, which concatenates the Top- k historical segments retrieved via dense retrieval with the query as context (Salemi and Zamani, 2025a).
- (3) **Plan-RAG:** A *Plan-Retrieve-Generate* strategy that uses a fine-tuned planning model to derive query-related aspects, providing explicit guidance alongside retrieved context (Salemi and Zamani, 2025a).
- (4) **Long-Context:** Concatenates raw user history in reverse chronological order, limiting the context window to 32k tokens for Qwen and 8,192 for Gemma-2 (Liu et al., 2024).
- (5) **Mem0:** An unstructured agent memory baseline utilizing incremental updates, which we mirror by directly evaluating our GME module outputs to ensure fair comparison (Chhikara et al., 2025).

5.4 Implementation Details

Model Configurations. We evaluate on Qwen-2.5-7B-Instruct², Gemma-2-9B-it³, and Qwen-3-30B-A3B-Instruct-2507⁴, deployed via vLLM (Kwon et al., 2023). We utilize Qwen-3-30B as the back-

¹<https://huggingface.co/Qwen/Qwen2.5-32B-Instruct>

²<https://huggingface.co/Qwen/Qwen2.5-7B-Instruct>

³<https://huggingface.co/google/gemma-2-9b-it>

⁴<https://huggingface.co/Qwen/Qwen3-30B-A3B-Instruct-2507>

bone for offline memory construction (Sec. 4.1–4.2). Memory extraction parameters are set to $m = 5$ (candidates), $s = 5$ (retrieval limit), and $m_s = 5$ (update interval). Thresholds are set to $\tau_m = 0.9$ (merge), $\tau_a = 0.7$ (disambiguation), and $\tau_p = 0.85$ (relation deduplication). During inference, we set the temperature to 0.1 and the max output length to 4096 tokens.

Retrieval and Storage. We employ Neo4j Community Edition⁵ for graph storage and Contriever ($d = 768$) (Izacard et al., 2021) for embeddings and retrieval. Baselines retrieve the top- $k = 10$ documents, whereas Mem-ABC retrieves the top- $t = 100$ memory triples to construct the subgraph context.

Baselines & Hardware. Baselines marked with (*) are reproduced using our backbone LLMs for fair comparison, strictly preserving their original hyperparameters. All experiments are conducted on two 80GB NVIDIA H100 GPUs.

5.5 Main Results

Table 1 presents the performance comparison between Mem-ABC and baseline methods across three domains: *Arts & Entertainment*, *Lifestyle & Personal Development*, and *Society & Culture*. The results demonstrate that Mem-ABC achieves superior performance across most datasets and backbone architectures.

Firstly, compared to the Non-Personalized baseline, all personalization methods exhibit significant performance gains. This fundamental improvement highlights the necessity of leveraging user history in PQA tasks.

Regarding retrieval and memory-based paradigms, Mem-ABC demonstrates a clear advantage, significantly outperforming baselines like RAG-Personalized and Plan-RAG (improving average accuracy on Qwen-2.5-7B-Instruct from 0.4282 to 0.5341). Furthermore, our framework surpasses the strong agentic memory baseline, Mem0. While Mem0 mitigates narrative noise through dynamic updates, its reliance on unstructured textual storage limits its ability to integrate fragmented clues scattered across interactions. In contrast, Mem-ABC resolves this semantic ambiguity by transforming raw text into structured attitude-aware representations.

Notably, Mem-ABC also outperforms the Long-Context baseline. While long-context models theo-

Table 1: Main results of accuracy comparison across three domains. The best performance is highlighted in **bold**, and the second best is underlined. *indicates reproduced results. †indicates statistical significance ($p < 0.05$) compared to the strongest reproduced baseline, calculated using the one-sided Wilcoxon signed-rank test on matched instance-level scores.

Method	Arts & Ent.	Life. & Dev.	Soc. & Cul.	Average
<i>Qwen-2.5-7B-Instruct</i>				
No-Personalized	0.3129	0.4582	0.4769	0.4160
RAG-Personalized	0.3397	0.4481	0.4967	0.4282
Plan-RAG	0.3518	0.4818	0.5240	0.4525
Long-Context*	<u>0.3873</u>	<u>0.5212</u>	<u>0.5375</u>	<u>0.4820</u>
Mem0*	0.3810	0.4963	0.5394	0.4722
Mem-ABC	0.4127†	0.5693†	0.6203†	0.5341†
<i>Gemma-2-9B-it</i>				
No-Personalized	0.1860	0.3858	0.4094	0.3271
RAG-Personalized	0.2929	0.4232	0.4834	0.3998
Plan-RAG	0.3548	0.4671	0.5481	0.4567
Long-Context*	0.2810	0.4510	0.4862	0.4061
Mem0*	0.2779	0.3485	0.4216	0.3493
Mem-ABC	<u>0.3279</u>	0.5102†	0.5614†	0.4665†
<i>Qwen3-30B-A3B-Instruct-2507</i>				
No-Personalized*	0.4034	0.5602	0.6005	0.5214
RAG-Personalized*	0.4644	0.6020	0.6526	0.5730
Plan-RAG*	0.4816	0.6217	0.6873	0.5969
Long-Context*	0.4788	<u>0.6468</u>	0.6983	<u>0.6080</u>
Mem0*	0.4477	0.6251	0.6374	0.5701
Mem-ABC	0.4939	0.6605†	0.7291†	0.6278†

retically access more history, they often suffer from narrative noise. Mem-ABC’s high-density memory structure effectively acts as a denoising filter, focusing on core user attitudes rather than the full, noisy context. In addition, both cross-model evaluation with Llama-3.1-70B and human preference evaluation further confirm the robustness of our results, where Mem-ABC preserves the best ranking under an independent evaluator and achieves a 56.3% win rate against Plan-RAG in human judgments (Appendices A.3 and B).

5.6 Ablation Study

To assess each component in Mem-ABC, we conducted an ablation study using the Qwen-2.5-7B-Instruct backbone. We analyzed the framework from two perspectives: (1) **Module-level**, evaluating the General (GME) and Attitude-Aware Memory Extraction (AME) modules; and (2) **Relation-level**, examining both objective factual connections and the subjective dimensions defined by the ABC Model. Results are summarized in Table 2.

Impact of Modules (GME & AME):

- **w/o GME:** Removing the General Memory Extraction module necessitates constructing memory directly from raw history. This leads to a performance decline from 0.5341 to 0.5126. As illustrated in Figure 3, GME effectively filters

⁵<https://neo4j.com/deployment/self-hosted/>

Table 2: Ablation study on Qwen-2.5-7B-Instruct. We analyze the impact of different components (GME, AME) and specific attitude relations (Affect, Behavior, Cognition). **Bold** indicates the best performance.

Model	Arts & Ent.	Lifestyle & Dev.	Society & Culture	Average
Ours (Mem-ABC)	0.4127	0.5693	0.6203	0.5341
<i>Relation-level Ablation</i>				
w/o Behavior	0.4086	0.5617	0.6067	0.5257
w/o Cognition	0.4043	0.5591	0.6088	0.5241
w/o Affect	0.4024	0.5523	0.6118	0.5222
w/o Objective	0.4055	0.5431	0.6159	0.5215
<i>Module-level Ablation</i>				
w/o GME	0.3909	0.5436	0.6034	0.5126
w/o AME	0.3810	0.4963	0.5394	0.4722
RAG-Personalized	0.3397	0.4481	0.4967	0.4282

out approximately 69.6% to 71.4% of redundant entities while retaining core semantic information. Without GME, the memory becomes bloated with narrative noise, which diminishes retrieval precision and increases storage overhead.

- **w/o AME:** Removing AME degrades the system to unstructured textual memory (retrieving the top-30 items from $\mathcal{M}_g$). Since GME adapts the Mem0 architecture, this variant mirrors the Mem0 baseline (Section 5.5). The sharp performance decline (to 0.4722) confirms that our primary gain stems from the structured *ABC Model* rather than simple text summarization.

Impact of Relation Types: As detailed in Table 2, removing any single relation type consistently impairs model performance, confirming that each dimension provides non-redundant, complementary signals. Notably, while the absence of Objective facts yields the largest performance drop (0.5215), validating their foundational role in context grounding, removing the subjective Affect dimension causes a nearly identical decline (0.5222). This reinforces our core philosophy: objective facts provide the indispensable base knowledge, whereas subjective attitudes act as the critical engine for inferring latent user intent. Ultimately, Mem-ABC’s success stems from the strong synergy between the *Tripartite Model* and objective facts, proving that a purely fact-centric memory is insufficient for deep personalization.

Notably, even the ablated variants (e.g., w/o GME or w/o Affect) still outperform the traditional RAG-Personalized baseline (0.4282). This validates the superiority of our agent memory mechanism over flat retrieval approaches.

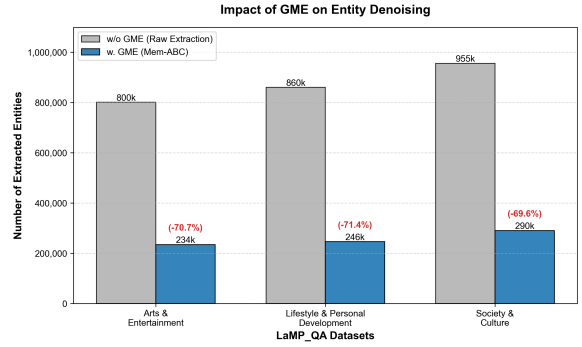


Figure 3: Comparison of extracted entity nodes between raw extraction (w/o GME) and our method (w. GME). The numbers in red indicate the **noise reduction rate**.

5.7 Token Efficiency and Computational Cost

During online inference, Mem-ABC significantly reduces context token consumption compared to baselines. By distilling noisy histories into structured attitude triples, it provides high information density, enabling the LLM to focus exclusively on core user preferences while maintaining low QA latency (averaging ~ 4 s).

Although constructing the memory graph incurs an offline computational overhead, it serves as a necessary, one-time preprocessing investment. In high-frequency interactive scenarios, the cumulative online savings in API costs and latency far outweigh this initial expense. A comprehensive cost and token efficiency analysis is detailed in Appendix E.

6 Conclusion

In this paper, we presented Mem-ABC, a novel agent memory framework that integrates the Tripartite Model of Attitudes to overcome the limitations of traditional RAG in personalization. Unlike passive text retrieval, our approach employs a two-stage offline mechanism—General Memory Extraction and Attitude-Aware Relation Extraction—to filter narrative noise and construct a structured memory explicitly modeling user subjectivity. Extensive experiments on the LaMP-QA benchmark demonstrate that Mem-ABC achieves state-of-the-art performance. This work validates the efficacy of grounding agent memory in psychological theories, offering a robust paradigm for developing next-generation agents with enhanced empathy and personalized understanding.

Limitations

The primary limitation of our work is the exclusive reliance on open-weight Large Language Models (e.g., Qwen-2.5, Gemma-2) for experimentation, omitting proprietary models such as GPT-4o or Claude-3.5. This decision was driven by two key factors. First, the extensive token consumption required for our two-stage offline memory construction and iterative ablation studies imposed significant financial and computational constraints.

Second, and more importantly, the substantial performance gains observed on open-source models already provide robust evidence for the efficacy of the proposed Mem-ABC framework. By validating our approach on accessible models, we demonstrate that the benefits of attitude-aware memory are structural rather than dependent on the latent knowledge of ultra-large-scale proprietary models, thereby facilitating easier reproduction by the research community.

References

- Steven J Breckler. 1984. Empirical validation of affect, behavior, and cognition as distinct components of attitude. *Journal of personality and social psychology*, 47(6):1191.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.
- Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka, Chien Van Nguyen, Thien Huu Nguyen, and Hamed Zamani. 2024. Longlamp: A benchmark for personalized long-form text generation. *arXiv preprint arXiv:2407.11016*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173.
- Reza Yousefi Maragheh, Pratheek Vadla, Priyank Gupta, Kai Zhao, Aysenur Inan, Kehui Yao, Jianpeng Xu, Praveen Kanumala, Jason Cho, and Sushant Kumar. 2025. Arag: Agentic retrieval augmented generation for personalized recommendation. *arXiv preprint arXiv:2506.21931*.
- Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*.
- Milton J Rosenberg. 1960. Cognitive, affective, and behavioral components of attitudes. *Attitude organization and change*, pages 1–14.
- Alireza Salemi, Cheng Li, Mingyang Zhang, Qiaozhu Mei, Zhuowan Li, Spurthi Amba Hombaiah, Weize Kong, Tao Chen, Hamed Zamani, and Michael Bendersky. 2025. Pathways of thoughts: Multi-directional thinking for long-form personalized question answering. *arXiv preprint arXiv:2509.19094*.
- Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. **LaMP: When large language models meet personalization**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.
- Alireza Salemi and Hamed Zamani. 2025a. **LaMP-QA: A benchmark for personalized long-form question answering**. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1159, Suzhou, China. Association for Computational Linguistics.
- Alireza Salemi and Hamed Zamani. 2025b. Learning from natural language feedback for personalized question answering. *arXiv preprint arXiv:2508.10695*.
- Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval augmented generation with collaborative filtering for personalized text generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1294–1304.
- Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister.

2025. 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)*, pages 8416–8439, Vienna, Austria. Association for Computational Linguistics.
- Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. 2025. Mem- α : Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents.
- Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. 2025. Memory-rl: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*.
- Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. 2025. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*.
- Mark P Zanna and John K Rempel. 1988. Attitudes: A new look at an old concept. *Psychology Press*.
- Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. 2025. G-memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*.

A Dataset and Evaluation Metrics

A.1 LaMP-QA Dataset

The experiments in this work are conducted on the LaMP-QA benchmark, a dataset specifically designed for evaluating personalized long-form question answering. The dataset is derived from the SE-PQA dataset, which aggregates data from the StackExchange platform, providing rich time-based interaction data.

The benchmark is categorized into three primary domains: (1) Arts & Entertainment, (2) Lifestyle & Personal Development, and (3) Society & Culture, encompassing over 45 subcategories. To ensure the dataset focuses on queries requiring personalization, factoid questions and questions lacking sufficient narrative detail were filtered out using large language models (LLMs). To further ensure the reliability of the personalized scoring rubrics, a human evaluation study was conducted on a sampled subset, achieving an average quality score of 4.9/5.

Regarding dataset statistics, the benchmark contains a total of 24,333 training, 2,603 validation, and 2,830 test samples. The specific distribution across splits (Train/Validation/Test) for each category is as follows: Arts & Entertainment (9349/801/767), Lifestyle & Personal Development (7370/892/989), and Society & Culture (7614/810/1074). Unless otherwise noted, all experimental results reported in this study are evaluated on the test set to ensure robust performance assessment.

Each instance in the dataset consists of the following components:

- **User Query** (x_u): The current question posed by the user.
- **User History** (H_u): A profile consisting of the user’s previously asked questions and their descriptions, which captures their information-seeking behavior and preferences.
- **Question Narrative** (r_{x_u}): A detailed description accompanying the question where users explicitly articulate their specific information needs and context.
- **Personalized Scoring Rubrics** (E_{x_u}): A set of high-quality aspects extracted from the question narrative that an ideal response should cover.

A.2 Evaluation Metric

To assess the quality of the generated responses, we adopt an aspect-based evaluation framework as proposed in LaMP-QA. For each query x_u , a set of personalized aspects $E_{x_u} = \{e_i\}_{i=1}^{n_{x_u}}$ is extracted from the user-provided narrative r_{x_u} , representing the specific expected content that an ideal response should cover.

The evaluation metric $\mu(x_u, \hat{y}_u, E_{x_u}, r_{x_u})$ measures the degree of alignment between the generated response and these expected aspects. Crucially, the aspect set E_{x_u} remains invisible to the model during the generation process. This ensures that the personalization capability is genuinely derived from the user history H_u , rather than achieved by directly matching the evaluation aspects.

The scoring process is conducted using an evaluator LLM (e.g., Qwen 2.5-32B) which rates the generated response against each aspect $e \in E_{x_u}$ on a 3-point scale:

- **Score 0:** The response does not cover the aspect.
- **Score 1:** The response somewhat covers the aspect.
- **Score 2:** The response covers the aspect very well.

The final score for a response is calculated by normalizing these scores (dividing by 2) and computing the average across all personalized aspects for the given query. This fine-grained evaluation method has been shown to achieve the highest alignment with human judgment compared to pairwise or direct scoring methods.

A.3 Reliability and Stability of the Evaluation Metric

To address potential concerns regarding the stability and alignment of LLM-based evaluations, we highlight the comprehensive reliability analyses conducted in the LaMP-QA benchmark, which strongly justify the adoption of this aspect-based metric.

Mitigating Position Bias and Instability: Alternative evaluation strategies, such as LLM-based pairwise preference scoring, have been shown to suffer from severe position bias. According to the benchmark’s analysis, simply reversing the presentation order of two candidate responses causes a 32B-parameter LLM to flip its preference in 78%

of cases. In contrast, the aspect-based evaluation assesses each personalized aspect independently, mitigating order sensitivity and providing a much more robust and objective measurement.

Alignment with Human Judgments: The adopted aspect-based evaluation strategy demonstrates a strong correlation with human preferences. In comparative validation studies, this fine-grained method achieved a 73% alignment rate with human annotators, significantly outperforming holistic direct scoring methods (which only achieved a 58% alignment rate). Furthermore, the human evaluators themselves demonstrated a high inter-annotator agreement (Cohen’s kappa of 0.726) when validating these preferences, confirming the grounded and objective nature of the extracted rubrics.

Effect of Evaluator Model Scale: The reliability of this metric is also highly dependent on the parameter scale of the evaluator LLM. Analyses indicate a clear correlation between model size and evaluation quality. Smaller models (e.g., 0.5B parameters) tend to inflate scores (averaging 0.94 out of 1.0) and behave as coarse binary classifiers, resulting in a poor human alignment rate of 48%. Conversely, larger-scale models, such as the Qwen 2.5-32B model utilized in our evaluation framework, possess the advanced reasoning capacity required to capture nuanced quality signals. They provide stricter, more realistic assessments (averaging 0.44) and achieve the highest alignment with human expectations. Additionally, generating evaluation scores with a low temperature setting (e.g., $\tau = 0.1$) further ensures the determinism and reproducibility of the reported results.

Addressing Model Family Bias: A potential concern in LLM-based evaluation is “model family bias,” where an evaluator LLM might implicitly favor text generated by models from its own architecture family. To rigorously verify that our performance gains stem from the proposed Mem-ABC architecture rather than evaluator bias, we conducted an additional cross-model meta-evaluation. Specifically, we employed **Llama-3.1-70B-Instruct** as an independent judge to re-assess the responses generated by our primary backbone model, Qwen-3-30B-A3B-Instruct. As shown in Table 3, while the absolute scores exhibit minor variations due to the distinct scoring calibrations of the Llama evaluator, **the overall performance trend remains broadly consistent with our main results: Mem-ABC still achieves the best performance across all three domains.**

Method	Art & Ent.	Life & Dev.	Society & Cul.
No-Personalized	0.3664	0.5526	0.6339
RAG-Personalized	0.4724	0.6405	0.7208
Long Context	0.4818	0.6768	0.7453
Plan-RAG	0.4931	0.6625	0.7367
Mem0	0.4694	0.6578	0.6922
Mem-ABC (Ours)	0.5026	0.6909	0.7857

Table 3: Cross-model evaluation results using **Llama-3.1-70B-Instruct** as an independent evaluator to assess outputs generated by the Qwen-3-30B backbone. The consistent relative rankings demonstrate the absence of model-family bias.

B Human Preference Evaluation

To further examine whether the automated aspect-based evaluation reflects human judgments, we conducted a complementary human preference study.

Experimental Setup. We randomly sampled 300 test instances from the LaMP-QA benchmark, with 100 instances from each domain. For each instance, two independent annotators were shown the user query, the query narrative, and two anonymized responses generated by Mem-ABC and the strongest baseline, Plan-RAG. The response order was randomized. Annotators were asked to choose the response that better matched the user’s implicit intent, provided more appropriate personalization, and remained factually consistent. They could also select a tie when the two responses were comparable.

Quantitative Results. As shown in Table 4, Mem-ABC is preferred over Plan-RAG across the three domains, which is consistent with the trend observed in the main results. The highest win rate appears in the *Society & Culture* domain, where Mem-ABC is preferred in 61.0% of the cases. The average Cohen’s Kappa between the two annotators is 0.754, indicating substantial agreement.

Alignment with LLM Evaluation. We further compare the human majority preference with the scoring direction given by the automated evaluator. The average alignment rate is 81.7%, with consistent decisions on 245 out of 300 instances. This result suggests that the aspect-based evaluator is reasonably aligned with human preference and can capture the personalization improvements introduced by Mem-ABC.

C Sensitivity Analysis

We further examine the robustness of our framework with respect to the retrieval hyperparameter t ,

Table 4: Human preference evaluation results (Mem-ABC vs. Plan-RAG) across 300 sampled instances, alongside inter-annotator agreement (Cohen’s κ) and LLM-Human alignment rates.

Domain ($N = 100$)	Win (%)	Tie (%)	Loss (%)	Cohen’s κ	LLM Align (%)
Arts & Entertainment	52	31	17	0.722	78
Lifestyle & Dev.	56	29	15	0.759	82
Society & Culture	61	26	13	0.781	85
Overall Average	56.3	28.7	15.0	0.754	81.7

which governs the maximum number of memory triples retrieved from the Attitude-Aware Memory. As illustrated in Figure 4, we vary t from 50 to 400 using the Qwen-2.5-7B-Instruct backbone.

The results indicate a rapid performance improvement as t increases from 50 to 100, suggesting that a sufficient volume of structured context is essential for accurately capturing user intent. However, beyond $t = 100$, the performance curves across all three domains tend to saturate. This implies that Mem-ABC effectively concentrates the most critical user information—both attitudes and facts—within the top-ranked relations. Consequently, we adopt $t = 100$ as the optimal balance between accuracy and computational efficiency.

D Reliability of ABC Attitude Extraction

To ensure our Tripartite Model of Attitudes (Affect, Behavior, Cognition) avoids hallucinations and reliably captures user profiles, we conducted a rigorous human evaluation.

Setup: We sampled 300 extracted relations (75 per category: Affective, Behavioral, Cognitive, and Objective). Three independent annotators reviewed them against historical contexts to verify their accuracy. Using a majority-vote criterion, the inter-annotator agreement reached a Fleiss’ Kappa of 0.82.

Results: As shown in Table 5, our framework achieved an overall accuracy of 91.33%. This high accuracy—especially in Affective and Behavioral dimensions ($>93\%$)—confirms the framework robustly extracts well-grounded user attitudes.

Category	Correct	Incorrect	Accuracy
Affective	71	4	94.67%
Behavioral	70	5	93.33%
Cognitive	66	9	88.00%
Objective	67	8	89.33%
Overall	274	26	91.33%

Table 5: Human evaluation of relation extraction reliability.

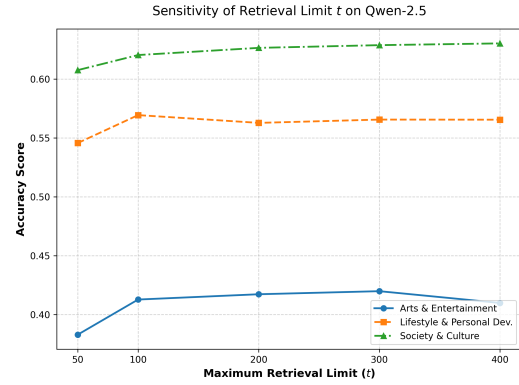


Figure 4: Sensitivity analysis of the **maximum retrieval limit** t across three domains. Experiments are conducted using **Qwen-2.5-7B-Instruct** as the backbone. Performance improves significantly as t increases to 100 and stabilizes thereafter, indicating semantic saturation.

E Token Efficiency and Computational Cost

To clarify the resource requirements and scalability of Mem-ABC, we conducted supplementary experiments by randomly sampling 300 users across the three datasets. These users were categorized into five groups (Bins) based on the count of their interaction history. The experiments were conducted with a concurrency of 1 on two NVIDIA H100 GPUs.

Table 6 reports the average end-to-end latency (both offline graph construction and online QA), token consumption, and API costs per user for each group. Furthermore, to evaluate context efficiency during the generation phase, Figure 5 compares the online token consumption of Mem-ABC against strong baselines. Note that for this online comparison, the reported input tokens strictly refer to the retrieved user context length. Across all experiments, all token statistics (both offline and online) are uniformly computed using the Contriever tokenizer to ensure a fair and consistent comparison.

Key Observations:

- **Practical Online Efficiency & Token Compactness:** In addition to achieving higher accuracy, the efficiency of context utilization is critical for practical LLM agents, as shorter contexts directly translate to reduced inference latency and lower API costs. As illustrated in Figure 5, Mem-ABC significantly reduces online token usage by approximately 50%–60% compared to traditional RAG. For instance, in the *Society & Culture*

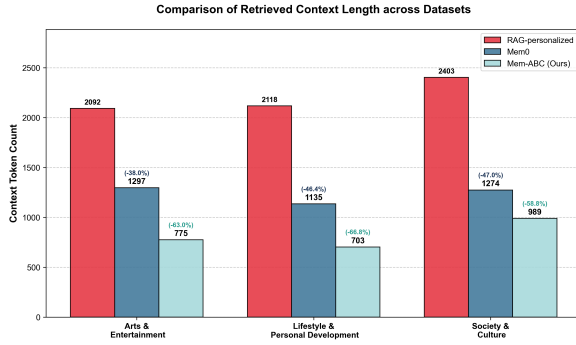


Figure 5: Comparison of average context token usage across different domains.

domain, RAG-Personalized consumes over 2,400 tokens per query due to the retrieval of noisy raw narratives. In contrast, Mem-ABC achieves superior accuracy using only 989 tokens. Even compared to the memory-based baseline Mem0, our method maintains a consistent advantage in context compactness. By distilling massive historical interactions into these highly structured triples, Mem-ABC maintains low online QA latency (averaging ~ 4 s) and keeps resource requirements for typical users within a highly practical range.

- **Offline vs. Online Trade-off:** For users with extensive histories (e.g., Bin 5), we acknowledge the computational overhead of offline memory construction. However, this is a one-time preprocessing step essential for capturing fine-grained user preferences over time. While the initial construction cost increases with history length, the cumulative savings in online inference costs in high-frequency interaction scenarios will far outweigh this initial offline overhead. Additionally, our current latency metrics are conservatively evaluated with a concurrency of 1. In real-world production environments, the offline processing time can be significantly reduced through request batching, asynchronous execution, and advanced LLM serving frameworks.

F Case Study

To qualitatively evaluate the impact of structured memory on handling user ambiguity, we analyze a representative case shown in Figure 6. The user poses an open-ended question: *What are some Household foods to mix with medicine?* While

the query is syntactically simple, the underlying narrative reveals a specific, unstated intent: the user needs high-palatability options to administer dewormer to **feral cats**.

Susceptibility to Narrative Noise. The **RAG-Personalized** baseline illustrates the detriment of *contextual distraction*. The retriever captures a historical segment regarding the user’s personal dysphagia (“*unable to swallow whole pills*”). Lacking structured disambiguation, the model erroneously anchors to this **irrelevant human-centric context**, restricting solutions to “Juice” or “Water.” These suggestions fail to align with the implicit animal-care intent, resulting in a low score of 0.125.

Efficacy of Attitude-Aware Modeling. In contrast, **Mem-ABC** correctly decodes the implicit intent by leveraging **structured user profiles**. By retrieving ABC-based triples such as (*User, has, dog*) and (*User, breeds, rabbit*), the system explicitly identifies the user’s role as an *animal caretaker*. This **structured attitude modeling** effectively filters out the noise, guiding the model to generate context-aware recommendations like “Peanut Butter” and “Yogurt”—options that align perfectly with the implicit narrative of medicating feral animals.

G Prompt Templates

To facilitate reproducibility and provide full implementation details, we present the core prompt templates used in the **Mem-ABC** framework. These prompts drive the two critical phases of our offline memory construction pipeline: **General Memory Extraction (Stage 1)** and **Attitude-Aware Memory Extraction (Stage 2)**.

For the **General Memory Extraction** stage, we illustrate the prompts for:

- **Candidate Memory Extraction** (Figure 7): Identifying and extracting concise memory fragments from raw user history.
- **Memory Updating** (Figure 8): The decision logic for the Dynamic Memory Decision Tool, enabling the agent to perform ADD, UPDATE, DELETE, or NOOP actions.
- **Summary Generation** (Figure 9): Integrating new insights into the global user summary.

For the **Attitude-Aware Memory Extraction** stage, we detail the prompts used to build the structured Attitude-Aware Memory:

Dataset	Module	Profile Len (Bin)	Avg Ext. Latency (s)	Avg QA Latency (s)	Avg In Tokens	Avg Out Tokens	Avg API Cost (\$)	Avg Profiles
Art & Ent.	GME	bin_1 (< 100)	186.95	/	295.2k	33.9k	0.045	48.80
		bin_2 (100-200)	465.27	/	865.4k	103.8k	0.133	134.00
		bin_3 (200-300)	879.19	/	1668.0k	198.8k	0.256	240.30
		bin_4 (300-400)	1232.88	/	2135.0k	246.9k	0.324	355.90
		bin_5 (400+)	2886.97	/	4964.9k	611.7k	0.769	668.40
	AME	bin_1 (< 100)	201.77	4.03	88.6k	34.2k	0.023	48.80
		bin_2 (100-200)	491.02	4.53	209.3k	79.7k	0.055	134.00
		bin_3 (200-300)	845.68	4.00	318.8k	125.6k	0.085	240.30
		bin_4 (300-400)	1466.51	4.03	428.6k	178.0k	0.118	355.90
		bin_5 (400+)	2892.96	4.14	892.8k	348.0k	0.237	668.40
Life. & Dev.	GME	bin_1 (< 100)	97.52	/	146.9k	16.1k	0.022	27.05
		bin_2 (100-200)	617.57	/	978.8k	116.2k	0.150	140.70
		bin_3 (200-300)	1107.58	/	1939.7k	239.1k	0.300	265.55
		bin_4 (300-400)	1960.75	/	2671.7k	322.6k	0.411	358.15
		bin_5 (400+)	6546.42	/	7576.9k	918.0k	1.167	1067.65
	AME	bin_1 (< 100)	56.43	3.85	36.2k	16.5k	0.011	27.05
		bin_2 (100-200)	467.56	4.28	178.0k	78.6k	0.051	140.70
		bin_3 (200-300)	957.94	4.41	262.7k	114.8k	0.075	265.55
		bin_4 (300-400)	1068.23	4.21	390.7k	178.1k	0.114	358.15
		bin_5 (400+)	3599.35	4.30	1055.3k	472.9k	0.306	1067.65
Soc. & Cul.	GME	bin_1 (< 100)	171.44	/	265.9k	31.4k	0.041	46.10
		bin_2 (100-200)	591.21	/	994.9k	123.5k	0.154	134.15
		bin_3 (200-300)	1365.03	/	1956.0k	238.5k	0.302	255.80
		bin_4 (300-400)	1699.18	/	2664.3k	339.4k	0.417	334.05
		bin_5 (400+)	6388.58	/	7443.2k	914.2k	1.151	1003.55
	AME	bin_1 (< 100)	162.35	4.09	61.7k	25.4k	0.017	46.10
		bin_2 (100-200)	491.63	4.50	160.4k	67.6k	0.045	134.15
		bin_3 (200-300)	914.80	4.28	274.2k	118.1k	0.077	255.80
		bin_4 (300-400)	974.58	4.22	280.6k	113.5k	0.076	334.05
		bin_5 (400+)	3268.45	4.04	955.3k	422.9k	0.275	1003.55

Table 6: Comprehensive scalability and computational cost analysis across three datasets, categorized by user profile count. The table reports offline extraction latency, online QA latency, token consumption, and API costs per user.

- **Entity Extraction** (Figure 10): Extracting and classifying entities into User, Semantic, and Episodic types.
- **Relation Extraction** (Figure 11): The critical instruction that guides the agent to infer *Affect*, *Behavior*, and *Cognition* relations based on the Tripartite Attitude Model.
- **Node Deduplication** (Figure 12): The adjudication logic for resolving entity ambiguity and merging duplicate nodes.
- **Personalized Answer Generation** (Figure 13): Synthesizing the user’s current question with retrieved memories to generate responses tailored to the user’s specific preferences and style.

Query Context	
Current Query: What are some Household foods to mix with medicine? Implicit Narrative: Trying to give some feral cats dewormer medication, and need something that a feral cat will love to eat. One cat liked yogurt, but not so much another one. Anything that all cats adore and is cheap and readily available, like peanut butter? Rubric Aspects: Palatability for feral cats, Attractiveness to all cats, Low cost and easy availability, Safety for use with medication.	
(a) Baseline: RAG-Personalized <Retrieved Context> <i>"I am completely unable to swallow whole pills... Is there any adult medication safe to chew? ...prescribed Advil."</i> <i>"Recently come into ownership of tough beef jerky... my teeth were painful..."</i> <Answer> Based on your past questions, it seems like you're always looking for solutions for pain... For medications like Advil, you can mix them with: 1. Juice: Help it go down easily. 2. Water: Simplest method. 3. Soft Foods: Applesauce... <Score>: 0.125 (Failed to address feral cat intent)	(b) Ours: Mem-ABC <Retrieved Context> (User, concerned_about, medication safety) [Affect] (User, has, 7-year-old large male dog) [Behavior] (User, breeds, new zealand white rabbit) [Behavior] (User, seeks, chewable medications) [Behavior] <Answer> Certainly! Based on your history, it seems you're looking for ways to administer medicine to your pets and even yourself... Here are household foods: 1. Peanut Butter: A small amount can make medicine palatable. 2. Yogurt or Pudding: Excellent for hiding texture. 3. Apple Sauce... <Score>: 0.625 (Successfully captures implicit intent)

Figure 6: **Qualitative Comparison on an Ambiguous Query.** (a) The **RAG baseline** suffers from contextual distraction, retrieving specific human health history (“swallowing pills”) and erroneously anchoring to a human-centric solution (Juice/Water). (b) **Mem-ABC** retrieves structured pet- and medication-related memories, which serve as indirect semantic anchors. This enables the model to disambiguate the intent and provide a comprehensive response covering the implicit target (feral animals) with options like Peanut Butter.

GENERAL MEMORY EXTRACTION PROMPT

You are a professional AI expert tasked with distilling a conversation into core memories. Your primary goal is to capture information that, if forgotten, would make future conversations feel disjointed, impersonal, or lacking in life details. Extract 1-5 pieces of information as appropriate.

[Context: Overall User Summary]

{global_summary}

[New Conversation Content]

"{profile_text}"

[Guiding Principles]

1. **Be Specific:** Extract precise facts. For example, extract "User likes jazz music," not "User likes music."
2. **Focus on Key Information:** Pay attention to preferences (likes/dislikes), goals, plans, personal experiences, habits, states, relationships, and key facts mentioned.
3. **Avoid Chitchat:** Do not extract generic greetings (e.g., "hello," "how are you?") or conversation fillers that lack meaningful content.
4. **Concise and Standalone:** Each memory must be a concise and standalone statement.
5. **Third-Person Perspective:** All memories must be written in third person (e.g., "User plans to...").
6. **Avoid Duplication:** If the new conversation content only repeats information already covered in the summary, extract an empty list.

[Output Format]

You must respond with a valid JSON object containing a single key "facts," where the value should be a list of strings.

[Example]

User Summary: "The user is a software developer."

New Conversation Content: "I finally booked my flight to Tokyo for next month's vacation! I'm so excited to try authentic ramen."

Your Output:

```
{
  "facts": [
    "User booked a flight to Tokyo for a vacation next month.",
    "User is excited to try authentic ramen in Tokyo."
  ]
}
```

Now, based on the provided summary and new conversation content, extract the core memories.

Figure 7: The prompt for extracting general memories from user history.

MEMORY DECISION PROMPT

You are a meticulous memory manager for an AI assistant. Your task is to analyze a new piece of information against a list of existing similar memories and decide on a precise action to maintain the consistency and accuracy of the knowledge base. Your comparison is not strictly based on the category, but rather on whether the facts overlap.

Action Guidelines:

- **ADD:** Choose this if the new information is entirely new and unrelated or not on the same level as any existing memory, with no overlap or contradiction.
- **UPDATE:** Choose this if the new information relates to an existing memory and adds crucial details, clarifies, or corrects it. The goal is to merge the new information and the most relevant existing memory into a single, more accurate memory.
- **DELETE:** Choose this only if the new information directly contradicts and invalidates an existing memory (e.g., a plan was canceled, or a preference has changed).
- **NOOP:** Choose this if the new information is an exact repetition, highly redundant to an existing memory, or trivial conversational filler, and provide a reason.

Existing Similar Memories:

{existing_memories}

New Information:

{new_fact}

Output Format Rules:

- If you choose **UPDATE**:
 - Line 1: UPDATE
 - Line 2: Write a single, concise, updated memory sentence that merges the new information with the most relevant old fact.
- If you choose **DELETE**:
 - Line 1: DELETE
 - Line 2: Provide a brief, one-sentence reason for the deletion.
- If you choose **ADD**:
 - Line 1: Simply output the single word ADD.
- If you choose **NOOP**:
 - Line 1: NOOP
 - Line 2: Provide a brief reason explaining why no action is taken.

Now, provide your decision for the new information below:

Figure 8: The prompt template for updating the general memory knowledge base.

SUMMARY UPDATE PROMPT

You are a professional user profiling expert. Your task is to generate an updated and more comprehensive user summary based on an existing summary and a recent raw conversation.

[Current User Summary]

{current_summary}

[Recent Raw Conversation]

{recent_raw_texts}

[Updating Requirements]

1. Read through the “Recent Raw Conversation” and extract relevant information.
2. Integrate these new details naturally into the “Current User Summary,” incorporating them into a cohesive and optimized summary.
3. Avoid simply listing the raw conversation as-is; instead, craft a fluid and descriptive summary.

[Output]

Provide the updated user summary in its entirety.

Figure 9: The instruction for updating the global user summary.

ENTITY EXTRACT PROMPT

System:

You are an expert Knowledge Graph (KG) Node Extractor.

Core Task

1. Analyze the user-provided INPUT_TEXT.
2. Identify all significant entities.
3. **Refine & Contextualize Names (CRITICAL):** - Do not simply copy raw substrings from the text. - **You MUST rewrite** entity names to be **specific, unambiguous, and self-explanatory** when viewed in isolation. - *Example:* Change “the paper” to “Research Paper on LLMs”; change “that tool” to “Jenkins”; change “the crash” to “System Outage Event”.
4. Classify them into **Instance** (specific things) or **Concept** (abstract ideas).
5. Always include the :User node.
6. Return *only* the JSON object.

Node Schema & Examples

- User: The user. (temp_id: “user”)
- Instance: A specific, unique, named entity. (e.g., “iPhone 15”, “Elden Ring”, “Shanghai”, “John Smith”, “that specific server crash”)
- Concept: An abstract category, domain, or idea. (e.g., “Gaming”, “Software Development”, “Anxiety”, “Cloud Computing”)

Constraints

1. The :User node must have temp_id: “user” and name: “User”.
2. All other nodes must have a unique temp_id (e.g., “n1”, “n2”...).
3. The name property must be the most representative phrase.
4. You must return only the JSON object.

EXPECTED_OUTPUT (JSON Example)

```
{
  "nodes": [
    { "temp_id": "user", "label": "User", "properties": { "name": "User" } },
    { "temp_id": "n1", "label": "Instance", "properties": { "name": "Elden Ring" } },
    { "temp_id": "n2", "label": "Concept", "properties": { "name": "Gaming" } },
    { "temp_id": "n3", "label": "Concept", "properties": { "name": "Music" } }
  ]
}
```

User:

INPUT_TEXT: {input_text}

OUTPUT:

Figure 10: The prompt used for entity extraction.

RELATION EXTRACT PROMPT

System:

You are a User Profiling Expert building a Knowledge Graph.

Core Goal

Extract facts that build a rich user profile, focusing on preferences, experiences, and beliefs.

Core Task

1. Analyze INPUT_TEXT using NODE_LIST_JSON.
2. Extract meaningful relationships.
3. Invent a concise invented_type (e.g., likes, visited).
4. Assign category (Affect, Behavior, Cognition, Objective).

Relation Schema

- source_temp_id, target_temp_id: node IDs.
- type: always "FACT".
- properties: { invented_type, category }

Constraints

1. Return only the JSON object.

EXPECTED_OUTPUT (JSON Example)

```
{
  "relations": [
    { "source_temp_id": "user", "target_temp_id": "n1", "type": "FACT",
      "properties": { "invented_type": "PLAYS", "category": "Behavior" } }
  ]
}
```

User:

INPUT_TEXT: {input_text}

NODE_LIST_JSON: {node_list_json}

OUTPUT:

Figure 11: The prompt used for relation extraction.

NODE DEDUPE PROMPT

System:

You are a Knowledge Graph Entity Linking Adjudicator.

Core Task

Based on the CONTEXT_TEXT, decide if the CANDIDATE_NODE is a synonym/abbreviation of the EXISTING_NODE (MERGE) or a completely new entity (NEW).

Decision Logic

1. MERGE: The CANDIDATE_NODE is a clear synonym, abbreviation, alias, or minor misspelling of the EXISTING_NODE. (e.g., "AirPort Capsule" vs "AirPort Time Capsule")
2. NEW: The entities are clearly different, or represent an instance/concept relationship. (e.g., "iPhone 15" vs "iPhone 14")

Constraints

1. You must return a single JSON object structured *exactly* like the EXPECTED_OUTPUT (JSON Example).
2. You must return *only* the JSON object, with no other text or explanation.

EXPECTED_OUTPUT (JSON Example)

Example 1 (Decision: MERGE)

```
{ "decision": "MERGE", "merge_target_uuid": "n1-uuid-existing", "reason": "The CANDIDATE 'Time Capsule' clearly refers to the EXISTING 'AirPort Time Capsule' in this context." }
```

Example 2 (Decision: NEW)

```
{ "decision": "NEW", "merge_target_uuid": null, "reason": "The CANDIDATE 'iPhone 15' is a distinct instance from the EXISTING 'iPhone 14'." }
```

User:

Input (Mimic the most relevant example)

CONTEXT_TEXT: "{input_text}"

EXISTING_NODE: "{existing_node}"

CANDIDATE_NODE: "{candidate_node}"

OUTPUT:

Figure 12: The prompt used for node deduplication.

PERSONALIZED QA PROMPT

System:

You are a helpful assistant designed to generate personalized responses to user questions.

Your input

- The user's current question from a post.
- The user's User-specific retrieved memories to learn about the user's preferences.

Example KG Memories

- user prefers evidence-based solutions
- user runs the dark eye
- the dark eye is_a tabletop role-playing game

Your task

- Answer the user's current question in a personalized way by considering their past posts and retrieved relevant user memories to understand their preferences.
- The retrieved user memories are potentially supplemental to the user's question; please carefully consider whether to integrate them into your response.
- Even if the retrieved content is not directly related to the current question, you can extract from it an understanding of the user's overall information, preferences, and style.

Your output

You should provide a detailed, helpful, and personalized answer to the current question.

User:

your input:

[Retrieved Relevant User Memories]

- {context}

[User's Question]

{question}

[Answer]

Figure 13: The prompt used for PQA generation.