VITABENCH 2.0: EVALUATING PERSONALIZED AND PROACTIVE AGENTS IN LONG-TERM USER INTERACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have evolved into interactive agents that collaborate with users in real-world tasks. Effective collaboration in such settings increasingly depends on understanding the user beyond what is explicitly stated, as user intent is often reflected in fragmented daily interactions and requires both personalized modeling and proactive interaction. However, existing personalized benchmarks do not yet fully capture how agents infer and act on evolving user preferences in everyday online service scenarios. To address this gap, we introduce VitaBench 2.0, a benchmark grounded in real-world service scenarios for evaluating personalized and proactive agents that serve the same user across sessions and over time. Motivated by recurring service needs, VitaBench 2.0 organizes tasks as temporally ordered sequences for individual users, whose preferences are reflected in fragmented and heterogeneous interactions. As time passes and user needs evolve, successful service naturally requires agents to extract, utilize, and update preferences based on newly available interactions. Some tasks also require proactiveness: when applying a preference depends on missing context, agents must obtain that information from users or the environment before acting. To study how agents retain relevant information across requests, we provide an extensible memory interface for controlled comparison of different memory architectures. We evaluate a diverse set of frontier proprietary and open-source LLMs and find that reliable personalized service remains challenging. Further analyses of memory use, preference-consistent execution, and proactive interaction identify recurring failure patterns, offering insights for improving agents that serve users over time.

1 INTRODUCTION

Recent advances in large language models (LLMs) have improved their capabilities in reasoning and tool use (DeepSeek-AI, 2025a;b; OpenAI, 2025b; Anthropic, 2025b), enabling them to evolve from passive text generators into interactive agents operating in real-world environments (Meituan LongCat Team, 2026; DeepSeek-AI, 2025c; Alibaba Cloud Community, 2025). Existing agent benchmarks assess how well agents complete single tasks through interaction with tools and environments (Jimenez et al., 2024; Liu et al., 2024; Zhou et al., 2024; Yao et al., 2024; Barres et al., 2025; He et al., 2025). As these agents move from individual tasks to sustained service for the same user, effective assistance depends on personalized understanding: decisions must remain aligned with the user's implicit needs (Liu et al., 2025; Zhang et al., 2024d).

However, existing personalized benchmarks do not yet fully capture the demands of everyday online service, where agents must execute tasks in complex environments while serving the same user across sessions and over time. Existing benchmarks assess personalized response generation (Salemi et al., 2024b; Kumar et al., 2024), evaluate implicit preference inference and the tracking of preference changes through multiple-choice questions (Jiang et al., 2025a;b; Li et al., 2026a), or examine proactive preference elicitation through user interaction (Li et al., 2025a). As time passes and user needs evolve, agents must update their understanding and apply current preferences when selecting among competing options and coordinating multi-step tool use in complex environments. Reliable

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.

execution also requires agents to assess whether the available information is sufficient for the task at hand. Evaluating such assistance therefore requires examining how personalization and proactiveness work together across successive service tasks.

Toward this end, we introduce VitaBench 2.0, an agent benchmark grounded in real-world service scenarios, where agents serve the same user as their needs and preferences evolve over time. These scenarios naturally support the evaluation of core personalization capabilities: (1) *preference extraction*, where agents infer implicit preferences from fragmented interactions; (2) *preference utilization*, where agents translate these preferences into user-specific decisions and actions in complex, realistic service environments; and (3) *preference updating*, where agents capture preference drift and revise their understanding as user behavior evolves. In these scenarios, successful completion of some tasks also calls for *proactiveness*: when a decision depends on context that is not yet available, agents must recognize the information gap and actively acquire the missing information before acting.

Following the general setup of existing agent benchmarks (Yao et al., 2024; Barres et al., 2025; He et al., 2025), we instantiate these real-world service scenarios in a simulated world where agents interact with users and execute actions through domain-specific tools. To capture recurring service needs, we organize each user's requests into temporally ordered task sequences spanning multiple domains. Between tasks, agents receive new dialogues and behavioral records containing fine-grained preference information, including evidence of preference changes. Memory plays a critical role in cross-session personalized service, enabling agents to retain and update user-specific knowledge across separate interactions. VitaBench 2.0 therefore provides an extensible memory interface that supports flexible implementations and controlled comparison across representative memory mechanisms (Chhikara et al., 2025; Xu et al., 2025; Yu et al., 2025).

We evaluate a diverse set of frontier proprietary and open-source LLMs and find that reliable personalized task completion remains challenging. Thinking generally improves personalization, but its gains vary across models and can reverse under some memory settings; memory itself does not consistently improve performance, with its effectiveness depending on the model and implementation. Diagnostic analyses further show that access to correct preferences does not ensure successful execution: agents may favor salient but unsuitable alternatives or fail to carry user constraints through tool use. When applying a preference depends on missing context, agents may similarly proceed without seeking clarification. These findings suggest that, despite advances in reasoning and memory, persistent gaps remain in reliable preference application and proactive information acquisition. VitaBench 2.0 provides a testbed for studying how these capabilities work together as agents serve the same user across sessions and over time.

2 Related Work

Personalized LLM. As large language models are increasingly deployed in user-facing applications, personalization has become a critical capability for aligning model outputs with individual user needs and preferences (Liu et al., 2025; Zhang et al., 2024d; Tseng et al., 2024). Achieving personalization requires models to capture user-specific characteristics and incorporate them into the generation process. Existing methods can be broadly understood from three alignment perspectives: input-level alignment, model-level alignment, and objective-level alignment. Input-level alignment enriches prompts with user-specific context. Retrieval-augmented methods obtain such context from interaction histories or external knowledge stores (Salemi et al., 2024a; Mysore et al., 2024; Richardson et al., 2023), while profile-based approaches explicitly summarize and inject user preferences into the prompt (Li et al., 2023a; Wu et al., 2024a). Model-level alignment adapts the model itself to generate outputs conditioned on user preferences, through parameter adaptation for white-box models (Tan et al., 2024; Zhang et al., 2024a) or model factorization frameworks for black-box models (Zhuang et al., 2024). Objective-level alignment incorporates personalization into training objectives, including personalized reward modeling (Jang et al., 2023), multi-objective preference optimization (Zhou et al., 2023), and causal preference modeling (Zhao et al., 2025b). As user interactions become increasingly long-term and informative, memory-augmented personalization has gained growing attention, supported by advances in memory systems and the increasing capability of LLMs to utilize them. This line of work augments LLMs with external memory mechanisms that support the storage, retrieval, and updating of user-relevant information over time (Packer et al., 2023; Chhikara et al., 2025; Xu et al., 2025). Complementary approaches optimize multi-turn collaboration (Wu et al.,

2025a) and support training and evaluation of proactive, personalized agents through user simulation and user-centric feedback (Sun et al., 2025b).

Benchmarks for LLM Personalization. As personalized LLMs become increasingly complex, there is a growing need for systematic evaluation benchmarks. Existing work can be broadly categorized along two dimensions: the form of user-specific information and the evaluation setting. From the input perspective, prior benchmarks assess personalization using various forms of user information, including explicit profiles (Zollo et al., 2024; Zhao et al., 2025a; Tan et al., 2025), user-authored documents (Salemi et al., 2024b; Kumar et al., 2024), and interaction histories with implicit or evolving preferences (Maharana et al., 2024; Wu et al., 2024b; Zhang et al., 2024c; Jiang et al., 2025a;b; Xiao et al., 2026; Li et al., 2025b; 2026a). From the evaluation perspective, early benchmarks mainly consider relatively static personalization scenarios, where user information is explicitly provided or derived from a fixed set of documents or profile attributes (Salemi et al., 2024b; Kumar et al., 2024; Zollo et al., 2024; Zhao et al., 2025a; Tan et al., 2025). More recent efforts place greater emphasis on long-term memory and dynamic user modeling, evaluating whether models can retain user-related information across extended interactions (Maharana et al., 2024; Wu et al., 2024b; Zhang et al., 2024c), infer implicit preference signals from conversational histories (Jiang et al., 2025b; Xiao et al., 2026; Li et al., 2025b), and adapt to preferences that evolve over time (Jiang et al., 2025a; Li et al., 2026a). Beyond using existing histories, interactive evaluations also examine how agents proactively elicit preferences in cold-start settings (Li et al., 2025a). VitaBench 2.0 complements these evaluations by grounding personalization in real-world online service scenarios where agents provide sustained service to users and must use inferred and updated preferences to guide executable actions in complex environments. Appendix B.2 compares these designs.

Benchmarks for LLM Agents. LLMs have evolved from text generators into autonomous agents capable of interacting with external tools and environments (Schick et al., 2023). Existing agent benchmarks have progressed from evaluating isolated tool-use capability to assessing increasingly realistic forms of interactive task execution. Early benchmarks mainly focus on API invocation and tool-use accuracy, evaluating whether models can select appropriate tools and generate valid arguments for a given user request (Li et al., 2023b; Patil et al., 2024). Subsequent benchmarks move toward more interactive and stateful settings, where agents must reason over multiple turns, track intermediate states, and respond to evolving context and feedback during execution (Farn & Shin, 2023; Wang et al., 2024a; Qin et al., 2023; Lu et al., 2024; Shi et al., 2026). More recent efforts further place agents in realistic execution environments, including web searching (Liu et al., 2024; Zhou et al., 2024), computer using (Xie et al., 2024), software engineering (Jimenez et al., 2024), and user-agent interaction (Trivedi et al., 2024; Yao et al., 2024; Barres et al., 2025; He et al., 2025; Li et al., 2026b; Wang et al., 2026b; Zheng et al., 2026), to evaluate end-to-end task completion under real-world constraints. This line of evaluation also encompasses preference elicitation and proactive interaction in executable mobile GUI environments (Chen et al., 2026). VitaBench 2.0 focuses on sustained service to the same user, requiring agents to apply evolving preferences across tasks and acquire missing context before acting.

3 VITABENCH 2.0

VitaBench 2.0 models sustained assistance in real-world service scenarios where user preferences are implicit and evolve over time. Agents infer preferences from fragmented interactions to guide decisions across the same user's requests. Figure 1 provides an overview. Each user u is associated with a profile P_u, evolving preferences $\mathcal{N}_u = (n_1, n_2, \ldots, n_L)$, and a temporal task sequence $\mathcal{T}_u = (t_1, t_2, \ldots, t_N)$, representing the user's changing needs across successive service requests. Between tasks t_{i-1} and t_i, the agent is exposed to newly introduced interaction histories that reflect emerging preferences or preference drift, and enabled to maintain a memory module $\mathcal{M}$ to store user information and support future decisions. We describe the task formulation and key benchmark modules below. Appendix C.2 analyzes the curated user profiles and preferences.

3.1 TASK SET

Tasks in VitaBench 2.0 are organized as temporally ordered sequences for individual users, where each sequence spans multiple domains. Each individual task t_i is an agentic task in which the

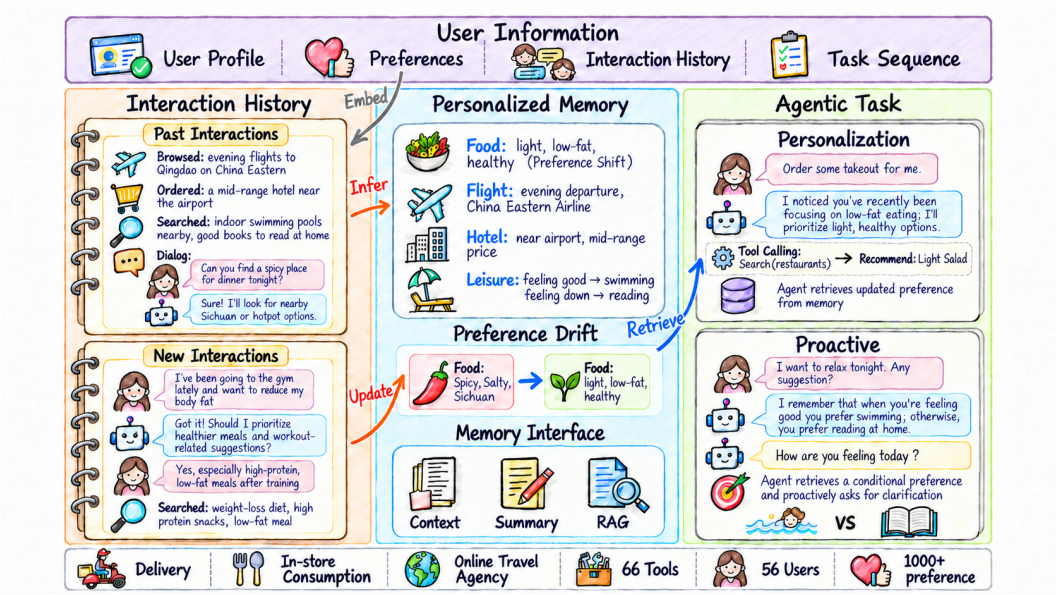

Figure 1: Overview of VitaBench 2.0. The agents are required to operate over temporal task sequences for each user, infer evolving user preferences from fragmented interactions, maintain these preferences via a memory mechanism, and make personalized and proactive decisions.

agent interacts with domain-specific tools and an executable environment to fulfill a user request. Concretely, each task can be modeled as a partially observable Markov decision process (POMDP):

$$\mathcal{P}_i = (\mathcal{S}_i, \mathcal{A}_i, \mathcal{O}_i, \mathcal{T}_i, r_i), \tag{1}$$

where $\mathcal{S}_i$ denotes the environment state, $\mathcal{A}_i$ the action space, $\mathcal{O}_i$ the observation space, $\mathcal{T}_i : \mathcal{S}_i \times \mathcal{A}_i \to \mathcal{S}_i$ the state transition function, and r_i the reward for task t_i.

Fulfilling these service requests requires both executing the appropriate operations and understanding the user: explicit constraints in the current query must be combined with implicit signals from fragmented past interactions. Specifically, a task instance is specified as:

$$t_i = (q_i, \mathcal{F}_i, \mathcal{E}_i, \mathcal{G}_i, \mathcal{H}_i), \tag{2}$$

where q_i is the user query, $\mathcal{F}_i$ is the set of available tools, $\mathcal{E}_i$ is the executable environment with underlying states, $\mathcal{G}_i$ is a set of evaluation rubrics, and $\mathcal{H}_i$ denotes the interaction histories exposed to agent between tasks t_{i-1} and t_i, simulating fragmented user interactions over time. Successful task execution requires the agent to identify user intent from q_i, select appropriate tools from $\mathcal{F}_i$, and infer relevant user preferences from $\mathcal{H}_{1:i}$ to make consistent and personalized decisions in $\mathcal{E}_i$.

When memory is enabled, the agent may update it using $\mathcal{H}_i$ before executing task t_i:

$$\mathcal{M}_i = \textsc{Update}(\mathcal{M}_{i-1}, \mathcal{H}_i). \tag{3}$$

At each step t within task t_i, the agent receives an observation $o_t \in \mathcal{O}_i$ consisting of the user query, dialogue history, and environment feedback from previous actions. The agent then selects an action conditioned on the current observation and updated memory state:

$$a_t \sim \pi(a_t \mid o_t, \mathcal{M}_i), \tag{4}$$

where the action space is given by

$$\mathcal{A}_i = \mathcal{A}_{\text{tool}} \cup \mathcal{A}_{\text{dialogue}}, \tag{5}$$

where $\mathcal{A}_{\text{tool}}$ denotes tool invocations and $\mathcal{A}_{\text{dialogue}}$ denotes natural-language responses to the user simulator. After executing a_t, the environment transitions to a new state s_{t+1} and returns a new observation o_{t+1}. The agent iterates between tool use and user interaction until the task is completed or a maximum number of steps is reached, producing a trajectory:

$$\tau_i = (o_0, a_0, o_1, a_1, \ldots, o_T, a_T). \tag{6}$$

An evaluator LLM assigns a reward by checking the trajectory τ_i and final action a_T against the atomic criteria in $\mathcal{G}_i$. These criteria specify the task requirements and user preferences that the completed operation must satisfy. Appendix A.3.4 details the evaluation protocol. Inheriting from VitaBench (He et al., 2025), we construct VitaBench 2.0 through systematic abstraction of real-world life-serving scenarios across three domains—Delivery, In-store Consumption, and Online Travel Agency—with a total of 66 tools. Detailed descriptions of the task pipeline and environment construction are provided in Appendix A.3 and Appendix A.2.

3.2 KEY MODULE

To ground sustained service in realistic user contexts, VitaBench 2.0 represents each user through a profile, evolving preferences, and fragmented interaction histories. We curate 56 users with more than 2,000 fine-grained preferences, covering diverse preference types and interaction contexts. All benchmark data are human-annotated. The data-driven construction of user profiles, preferences, and interactions draws on real-world service scenarios to reflect preference diversity and behavioral heterogeneity. Serving the same user across sessions also calls for retaining and updating relevant information, which we support through an extensible memory interface. We next describe the features of these components in detail.

User Profiles. Each user u is associated with a manually curated profile P_u, constructed in a data-driven manner to reflect realistic user characteristics. To ensure both diversity and realism in the user population, we model users along multiple dimensions, including demographics, geographic and socioeconomic attributes, occupation, and social context, with distributions aligned to real-world scenario statistics. A comprehensive analysis of the curated profiles is provided in Appendix C.2.1.

User Preferences. Each user u is also associated with a set of preferences $\mathcal{N}_u = \{n_1, n_2, \ldots, n_L\}$, spanning multiple aspects of daily life (e.g., dining, leisure and entertainment, shopping, travel, hobbies, and lifestyle habits). Preferences are expressed as natural language statements grounded in the user profile (e.g., "avoids spicy food due to a stomach condition"). As time passes, changes in a user's needs and circumstances can make earlier preferences outdated. We capture this evolution through fragmented user interactions that reflect preference drift over time. Between selected consecutive tasks, a subset of preferences may undergo one of three changes: (1) *addition*, where a new preference emerges; (2) *deletion*, where an existing preference becomes inactive; and (3) *modification*, where an existing preference shifts. In total, we manually curate 56 users with over 2,000 unique preferences. Detailed descriptions and illustrative examples are provided in Appendix A.1. A comprehensive analysis of the curated preference is provided in Appendix C.2.2.

Interaction History. In everyday service, user preferences are often reflected implicitly in fragmented interactions. We therefore encode preference evidence in fragmented interaction histories comprising user dialogues and behavioral records. As the agent progresses from task t_{i-1} to t_i, it is exposed to newly introduced interaction histories $\mathcal{H}_i$, which may reflect changes in the user's underlying preferences. Inspired by information accessibility in real-world scenarios, $\mathcal{H}_i$ contains two types of records: (1) *dialogues*, consisting of multi-turn user–agent conversations; and (2) *behaviors*, consisting of user behavior logs such as browsing, ordering, reviewing, and searching histories. Among these, not all interactions are preference-relevant. Instead, $\mathcal{H}_i$ can be viewed as comprising both signal interactions that reflect the user's underlying preferences and noise interactions that are irrelevant, ambiguous, or contextually misleading. Agents must therefore distinguish preference evidence from incidental behavior and use it to guide subsequent service decisions. Detailed construction of interaction history and illustrative examples are provided in Appendix A.1.

Memory Module. Effective cross-session service requires retaining user-specific knowledge beyond the current interaction. We therefore allow agents to maintain an external memory module $\mathcal{M}$ that carries relevant information into subsequent tasks. During task execution, agents with memory enabled access user history through this module rather than reading the full histories directly. Formally, before executing each task t_i, the agent is exposed to any newly available interaction history H_i and updates its memory:

$$\mathcal{M}_i = \textsc{Update}(\mathcal{M}_{i-1}, H_i). \tag{7}$$

During task execution, the agent conditions its actions on both the current observation and memory:

$$a_t \sim \pi(a_t \mid \textsc{Retrieve}(\mathcal{M}_i, q_i), o_t), \tag{8}$$

where $\textsc{Retrieve}(\mathcal{M}_i, q_i)$ returns task-relevant information from memory.

VitaBench 2.0 exposes these operations through an extensible interface for comparing memory mechanisms in cross-session service. We implement two representative mechanisms:

- **Agentic Memory.** The agent maintains a structured representation of user information and actively controls the memory content by deciding what information to retain, update, or discard. The memory is incrementally updated with each new history batch, and RETRIEVE returns all or a selective memory representation. This design requires the agent to perform selective abstraction, resolve conflicts across observations, and maintain long-term consistency.
- **RAG Memory.** Interaction records are stored in a memory bank with vector embeddings. UPDATE indexes new records, and RETRIEVE performs similarity-based retrieval given the task query. This design follows a fixed pipeline, where memory access is determined by retrieval without explicit control over what information is retained or discarded.

We provide a detailed discussion of memory mechanisms for agent systems in Appendix B.1.

3.3 PROACTIVENESS

In real-world service, knowing a user's preferences may not be sufficient to decide what to do, because their applicability can depend on the current circumstances. For example, if a user's preferred caffeine level depends on the time of day, ordering coffee for a meeting requires knowing when the meeting takes place. When this context cannot be inferred from the current query, history, or memory, the agent must recognize the information gap and seek user clarification or explore the environment.

We model these situations through tasks involving conditional preferences and missing but necessary context. Such tasks are interleaved with standard personalization tasks, requiring the agent to decide when additional information is needed before acting. For clarification, the simulator provides task-specific context when appropriately asked, without directly revealing the preferences that govern the decision. Agents must combine this context with previously inferred preferences to complete the task. Appendix A.3.2 details the simulator and its information-disclosure protocol.

4 EXPERIMENT

4.1 EXPERIMENTAL SETUPS

Models. We evaluate a diverse set of state-of-the-art proprietary and open LLMs, covering both non-thinking and thinking configurations when available. The evaluated models include OpenAI family, including GPT-3.5-Turbo, GPT-4o-mini, GPT-5, and o-series models such as o3 and o4-mini (OpenAI, 2025a;b;c;d;e); the DeepSeek family, including DeepSeek-R1 and DeepSeek-V4 variants (DeepSeek-AI, 2025b; 2026); Anthropic's Claude series, including Claude Sonnet and Claude Opus variants (Anthropic, 2025a;b; 2026); Google's Gemini series, including Gemini-2.5-Flash and Gemini-2.5-Pro (Comanici et al., 2025; Google DeepMind, 2025b;a); Qwen3-Max (Alibaba Cloud Community, 2025); GLM variants, including GLM-4.5, GLM-4.6, and GLM-5.1 (GLM-4.5 Team, 2025; Z.ai, 2025; 2026); ByteDance Seed series, including Seed-1.6 and Seed-2.0-Pro (ByteDance Seed, 2025; 2026); Kimi-K2.6 (Moonshot AI, 2026); LongCat-Flash (Meituan LongCat Team, 2025); and MiniMax-M2.7 (MiniMax, 2026). To ensure fair comparison, we distinguish between reasoning-enhanced (thinking) and non-reasoning (non-thinking) models. For hybrid architectures that support both modes, we evaluate the think-on and think-off configurations separately. We exclude smaller models due to the benchmark's difficulty and report thinking and non-thinking results separately.

Implementations. All agents are implemented as function-calling agents based on the OpenAI tool schema. A task ends when the agent emits `###STOP###`, execution fails, or the limit of 200 agent steps is reached. We use `gpt-4.1` as the user simulator and Claude-Opus-4.6 as the evaluator, with validation of the evaluator and diagnostic models in Appendix C.5. Each model and memory configuration is evaluated four times. The evaluator uses temperature 0 where supported; the task

Table 1: Performance of non-thinking and thinking models under different memory settings. The leaderboard is sorted by the Avg@4 on *Full Context* setting. The best performance is in bold.

Models	Full Context			Agentic Memory			RAG Memory		
	Avg@4	Pass@4	Pass^4	Avg@4	Pass@4	Pass^4	Avg@4	Pass@4	Pass^4
Non-thinking Models									
GPT-4o-mini (w/o thinking)	0.067	0.180	0.006	0.084	0.230	0.008	0.094	0.227	0.010
GPT-3.5-Turbo (w/o thinking)	0.140	0.314	0.019	0.231	0.467	0.056	0.205	0.409	0.058
LongCat-Flash-Chat (w/o thinking)	0.298	0.510	0.123	0.302	0.537	0.105	0.290	0.471	0.136
GLM-4.5 (w/o thinking)	0.307	0.529	0.127	0.330	0.569	0.112	0.316	0.523	0.152
Doubao-Seed-1.6 (w/o thinking)	0.326	0.512	0.171	0.340	0.576	0.128	0.351	0.543	0.174
GLM-4.6 (w/o thinking)	0.342	0.612	0.113	0.336	0.623	0.084	0.317	0.555	0.123
Kimi-K2.6 (w/o thinking)	0.378	0.632	0.147	0.397	0.674	0.145	0.383	0.621	0.163
GLM-5.1 (w/o thinking)	0.420	0.654	0.204	0.423	0.664	0.182	0.383	0.585	0.200
Doubao-Seed-2.0-pro (w/o thinking)	0.428	0.649	0.218	0.426	0.665	0.198	0.406	0.625	0.208
DeepSeek-V4-Pro (w/o thinking)	0.456	0.652	0.267	0.427	0.658	0.208	0.424	0.617	0.246
Thinking Models									
o4-mini (w/ thinking)	0.210	0.433	0.047	0.270	0.533	0.073	0.261	0.451	0.091
Gemini-2.5-Flash (w/ thinking)	0.282	0.556	0.064	0.312	0.567	0.099	0.309	0.543	0.106
Qwen3-Max (w/ thinking)	0.284	0.499	0.105	0.324	0.599	0.091	0.315	0.519	0.134
Gemini-2.5-Pro (w/ thinking)	0.331	0.604	0.109	0.378	0.638	0.137	0.320	0.578	0.109
MiniMax-M2.7 (w/ thinking)	0.349	0.585	0.136	0.376	0.629	0.150	0.335	0.534	0.148
GLM-4.6 (w/ thinking)	0.359	0.612	0.115	0.351	0.625	0.106	0.336	0.575	0.135
GLM-4.5 (w/ thinking)	0.369	0.620	0.157	0.330	0.581	0.114	0.343	0.559	0.160
Doubao-Seed-1.6 (w/ thinking)	0.373	0.599	0.176	0.383	0.646	0.123	0.375	0.591	0.179
DeepSeek-R1-0528 (w/ thinking)	0.396	**0.691**	0.131	0.412	**0.712**	0.118	0.390	**0.643**	0.153
o3 (w/ thinking)	0.403	0.652	0.169	0.401	0.669	0.154	0.362	0.588	0.158
Claude-4.5-Sonnet (w/ thinking)	0.417	0.658	0.197	0.397	0.642	0.178	0.374	0.573	0.185
GPT-5 (w/ thinking)	0.441	0.658	0.226	0.421	0.647	0.204	0.410	0.591	0.236
GLM-5.1 (w/ thinking)	0.450	0.650	0.270	0.453	0.680	0.239	0.383	0.540	0.226
DeepSeek-V4-Pro (w/ thinking)	0.472	0.649	0.294	0.449	0.656	0.256	0.430	0.584	0.271
Doubao-Seed-2.0-pro (w/ thinking)	0.474	0.684	0.270	0.428	0.650	0.224	0.339	0.495	0.205
Kimi-K2.6 (w/ thinking)	0.481	0.685	0.266	0.450	0.696	0.223	**0.434**	0.611	0.253
Claude-Opus-4.6 (w/ thinking)	**0.503**	0.664	**0.337**	**0.454**	0.645	**0.259**	0.430	0.565	**0.300**

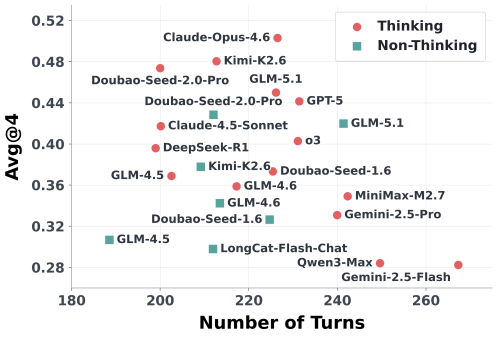

Figure 2: Average performance versus number of turns across models under full-context setting.

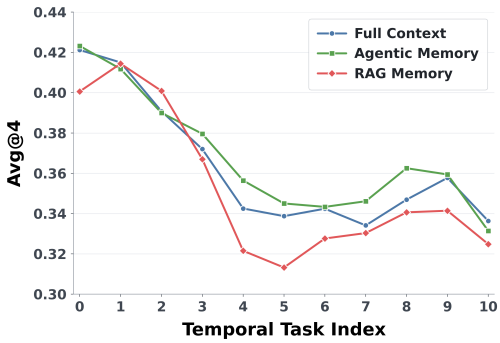

Figure 3: Average performance by task index.

agent and user simulator use their providers' default decoding settings. The runtime prompts appear in Appendix D.2. For memory mechanisms, we compare MemAgent (Yu et al., 2025) as agentic memory and the traditional RAG system as agent RAG memory. For detailed implementation and configuration settings, please refer to Appendix A.4.

Metrics. We report Avg@4, Pass@4, and Pass^4, computed from four independent runs and averaged over all tasks. Avg@4 measures the mean outcome reward across the four runs. Pass@4 denotes the probability that at least one of the 4 i.i.d. trials successfully completes the task. Pass^4 represents the probability that all 4 i.i.d. trials are successful.

4.2 Main Results

Table 1 presents evaluation results on VitaBench 2.0. We have the following observations.

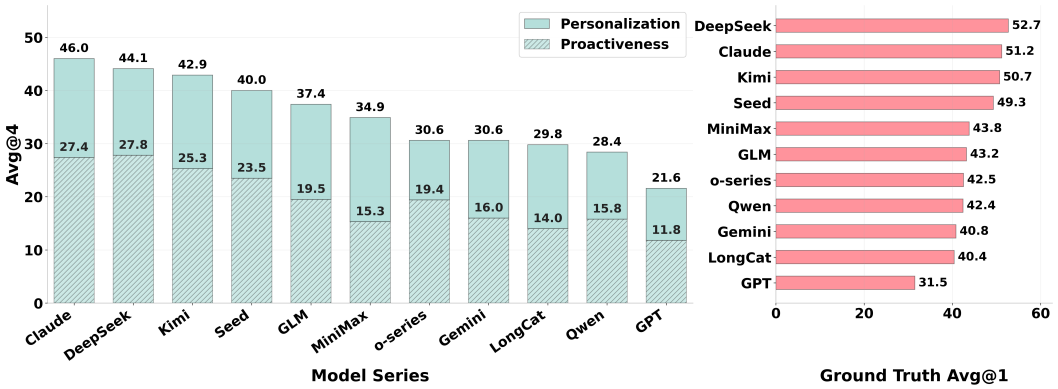

Figure 4: Analysis of model behavior on VitaBench 2.0. **Left**: average performance on proactive tasks across model series. **Right**: performance on VitaBench 2.0 given ground-truth user preferences.

Real-world personalization tasks remain highly challenging for current agents. Even under the *Full Context* setting where full interaction history is accessible, state-of-the-art models achieve only Avg@4 of around 0.5 and Pass^4 of around 0.3. Current agents thus struggle to infer and apply user preferences reliably across multi-step service tasks. As LLM agents enter real-world user-facing applications, this gap highlights a fundamental limitation in personalized decision-making.

Memory mechanisms play a critical role but do not always improve personalization performance. In realistic scenarios, interactions are often long-term and fragmented across sessions, making memory mechanisms essential for maintaining user representations. The benefit, however, depends on both the model and the memory implementation. Neither Agentic Memory nor RAG Memory consistently outperforms *Full Context*, and the same model can perform substantially differently across these two memory mechanisms. These differences suggest that effective memory requires more than retaining user information: its organization and retrieval must make relevant preferences accessible for downstream decisions. Memory design should therefore be evaluated jointly with the agent's ability to interpret and apply the stored information.

Reasoning enhancements generally improve personalization, but their benefits vary across models and memory settings. Table 1 shows that thinking improves Avg@4 in all seven matched-model comparisons under *Full Context*. GLM-4.6 improves from 0.342 to 0.359 and DeepSeek-V4-Pro from 0.456 to 0.472, while Kimi-K2.6 gains more substantially, from 0.378 to 0.481. The effect is less consistent with memory: Doubao-Seed-2.0-pro drops from 0.406 to 0.339 under RAG Memory. These differences suggest that reasoning benefits also depend on the available preference evidence. Compressed summaries or retrieved records may omit relevant constraints or retain outdated preferences, limiting what additional reasoning can achieve. Stronger reasoning thus needs to work alongside accurate preference extraction, updating, and use.

4.3 Analysis & Discussion

Accumulated long-term user interactions pose a fundamental challenge for context handling and memory management. We track mean Avg@4 across users and evaluated models at each task index, restricting the analysis to the first 10 positions, which are available for every user. Across complete sequences, median history length grows from 31,349 tokens at the first task to 133,666 at the final task, with some histories approaching 200K tokens. Figure 3 shows lower performance at later indices under both full-context and memory-based settings. Longer histories also bring irrelevant or outdated evidence and changing preferences, while the tasks themselves differ across positions. The trend therefore reflects these combined challenges of long-term user modeling. Appendix C.2.3 further analyzes the interaction histories.

Recognizing information gaps and proactively acquiring missing context remain challenging for current agents. Proactive tasks require agents to obtain missing context from users or the

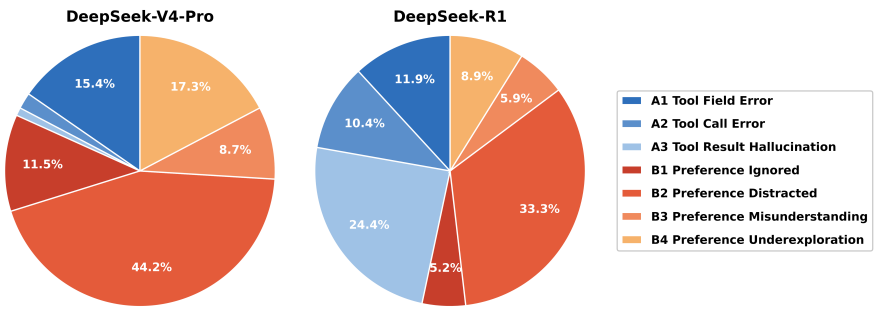

Figure 5: Failure pattern statistics for DeepSeek-V4-Pro and DeepSeek-R1. Category A denotes tool-related errors, category B denotes preference-related errors.

environment before deciding how to act. As shown in the left panel of Figure 4, proactive scores are lower than personalization scores across all model families: Claude scores 27.4 versus 46.0, DeepSeek 27.8 versus 44.1, and GLM 19.3 versus 36.4. The examined trajectories help explain this gap: agents sometimes proceed with incomplete information without initiating clarification. Recovering a conditional preference is therefore only part of the challenge; agents must also identify and obtain the missing context before acting. Appendix C.4.2 examines these stages.

Even given ground-truth user preferences, effectively leveraging them remains challenging for current agents. Providing ground-truth preferences for each task removes the need to recover them from history. The right panel of Figure 4 shows improved but still limited performance: DeepSeek and Claude score 52.7 and 51.2, while Seed, Kimi, and MiniMax reach 49.3, 43.8, and 42.6. Correct preferences alone are therefore insufficient for reliable execution. Agents must still interpret tool outputs and compare candidates, including highly rated or popular options that may violate user-specific constraints. These distractors can draw attention away from the preferences that should guide the decision. Effective utilization requires carrying those constraints through candidate selection and task execution. Appendix C.3 complements this analysis with extraction and updating measurements.

Failure pattern analysis shows that personalization emerges as the primary bottleneck in agent performance. Figure 5 summarizes the failure categories identified by Claude-Opus-4.6 in the full trajectories of DeepSeek-V4-Pro and DeepSeek-R1. Preference-related errors dominate the analyzed failures: agents miss relevant preferences or neglect them when choosing among available options, often favoring ratings or popularity. The stronger model exhibits fewer basic tool-use errors, such as improper tool invocations or incorrectly formatted arguments, yet preference violations across successive tool calls remain a major source of failure. Preference-consistent execution, a higher-level tool-use capability, remains a key bottleneck. Appendices C.4.1 and C.5.2 describe the taxonomy and its validation; Appendix D.1 illustrates how these errors affect a concrete decision.

5 Conclusion

We introduce VitaBench 2.0, an agent benchmark grounded in real-world service scenarios and the demands of repeatedly assisting the same user. Temporal task sequences, fragmented histories, evolving preferences, and conditional information needs naturally support joint evaluation of preference extraction, utilization, updating, and proactiveness through task completion. The extensible memory interface supports studying how agents retain and apply user information across sessions. Through extensive experiments on a diverse set of frontier models, we find that current LLM agents struggle to reliably infer, utilize, and update user preferences, especially when preferences evolve or when information is incomplete. Further analysis provides insights into the failure modes of current agents and the difficulties of long-term preference modeling. VitaBench 2.0 provides a testbed for advancing research on memory, personalization, and proactive behavior.

LIMITATION

VitaBench 2.0 focuses on personalized and proactive task execution as agents serve the same user across sessions and over time. Although its human-annotated user profiles, preferences, and interactions are informed by real-world distributions, they cannot fully capture the diversity and complexity of real-world users and service contexts. Generalization to other populations, domains, and live users requires further validation. Our evaluation spans multiple model families, sizes, and reasoning settings, but cost constraints leave broader model coverage to future work. To support realistic cross-session service, we provide an extensible memory interface with two representative implementations; evaluating a wider range of memory methods remains future work. Given space constraints, we provide additional benchmark construction details, implementation settings, evaluation protocols, diagnostic analyses, and case studies in the appendix.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and editing and to support the professional data team's benchmark annotation workflow. The data team remained responsible for annotation decisions and quality review. As described in Appendix A.4, GPT-4.1 serves as the user simulator and Claude-Opus-4.6 as the rubric-based evaluator. LLMs also support failure labeling and semantic matching between extracted preferences and human annotations; these uses are validated against human judgments in Appendix C.5. The evaluated agents and agentic memory mechanisms constitute additional experimental uses of LLMs described in the paper. The authors take responsibility for the final manuscript, benchmark content, and reported findings.

ETHICS STATEMENT

VitaBench 2.0 evaluates personalized and proactive agents in constructed service environments. User profiles, preferences, interaction histories, and environment records are manually designed with reference to aggregate application statistics; the benchmark does not contain private user records or identifiable personal data. The professional data team reviews these materials for sensitive content, unintended biases, and consistency, as described in Appendix A.5. Personalization nevertheless carries risks of incorrect preference inference, behavioral manipulation, and reinforcement of existing biases. We therefore emphasize transparency, user control over stored information and inferred preferences, and opportunities to correct or override agent decisions in downstream applications. Benchmark performance alone does not establish suitability for deployment with real users.

REPRODUCIBILITY STATEMENT

Section 3 defines the benchmark formulation and its personalization, memory, and proactiveness components. Appendix A details the construction of user profiles, preferences, histories, tasks, environments, and evaluation rubrics. Appendix A.4 specifies the memory backends, runtime models, decoding settings, execution limits, and repeated-evaluation protocol, while Appendix D.2 provides the prompt templates. Evaluator validation, variability across runs, and computational requirements are reported in Appendices C.5, C.6, and C.1, respectively. These descriptions document the benchmark construction and reference configurations used for the reported results.

REFERENCES

Alibaba Cloud Community. Qwen3-Max: Just Scale it, 2025. URL https://www.alibabacloud.com/blog/qwen3-max-just-scale-it_602621.

Anthropic. Introducing Claude 4, 2025a. URL https://www.anthropic.com/news/claude-4.

Anthropic. Introducing Claude Sonnet 4.5, 2025b. URL https://www.anthropic.com/news/claude-sonnet-4-5.

Anthropic. System Card: Claude Opus 4.6, 2026. URL https://www.anthropic.com/claude-opus-4-6-system-card.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating Conversational Agents in a Dual-Control Environment. *arXiv preprint arXiv:2506.07982*, 2025. URL https://arxiv.org/abs/2506.07982.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning (ICML)*, 2022. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

ByteDance Seed. Introduction to Techniques Used in Seed1.6, 2025. URL https://seed.bytedance.com/en/blog/seed1-6-%E7%B3%BB%E5%88%97%E6%A8%A1%E5%9E%8B%E6%8A%80%E6%9C%AF%E4%BB%8B%E7%BB%8D.

ByteDance Seed. Seed2.0 Model Card: Towards Intelligence Frontier for Real-World Complexity, 2026. URL https://lf3-static.bytednsdoc.com/obj/eden-cn/lapzild-tss/ljhwZthlaukjlkulzlp/seed2/0214/Seed2.0%20Model%20Card.pdf.

Tongbo Chen, Zhengxi Lu, Zhan Xu, Guocheng Shao, Shaohan Zhao, Fei Tang, Yong Du, Kaitao Song, Yizhou Liu, Yuchen Yan, et al. KnowU-Bench: Towards Interactive, Proactive, and Personalized Mobile Agent Evaluation. *arXiv preprint arXiv:2604.08455*, 2026. URL https://arxiv.org/abs/2604.08455.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. *arXiv preprint arXiv:2504.19413*, 2025. URL https://arxiv.org/abs/2504.19413.

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. URL https://arxiv.org/abs/2507.06261.

DeepSeek-AI. DeepSeek-V3.1 Model Card, 2025a. URL https://huggingface.co/deepseek-ai/DeepSeek-V3.1.

DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025b. URL https://arxiv.org/abs/2501.12948.

DeepSeek-AI. DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models. *arXiv preprint arXiv:2512.02556*, 2025c. URL https://arxiv.org/abs/2512.02556.

DeepSeek-AI. DeepSeek-V4-Pro Model Card, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130*, 2024. URL https://arxiv.org/abs/2404.16130.

Nicholas Farn and Richard Shin. ToolTalk: Evaluating Tool-Usage in a Conversational Setting. *arXiv preprint arXiv:2311.10775*, 2023. URL https://arxiv.org/abs/2311.10775.

Zafeirios Fountas, Martin A Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou-Ammar, and Jun Wang. Human-inspired Episodic Memory for Infinite Context LLMs. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2407.09450.

GLM-4.5 Team. GLM-4.5: Agentic, Reasoning, and Coding (ARC) Foundation Models. *arXiv preprint arXiv:2508.06471*, 2025. URL https://arxiv.org/abs/2508.06471.

Google DeepMind. Gemini 2.5 Flash and Native Capabilities (Audio & Image) Model Card, 2025a. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Flash-Model-Card.pdf.

Google DeepMind. Gemini 2.5 Pro Model Card, 2025b. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Pro-Model-Card.pdf.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv preprint arXiv:2410.05779*, 2024. URL https://arxiv.org/abs/2410.05779.

Bernal J. Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. In *Advances in Neural Information Processing Systems*, pp. 59532–59569. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-1902. URL https://papers.nips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2502.14802.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval Augmented Language Model Pre-Training. In *International Conference on Machine Learning (ICML)*, 2020. URL https://proceedings.mlr.press/v119/guu20a.html.

Wei He, Yueqing Sun, Hongyan Hao, Xueyuan Hao, Zhikang Xia, Qi Gu, Chengcheng Han, Dengchang Zhao, Hui Su, Kefeng Zhang, et al. VitaBench: Benchmarking LLM Agents with Versatile Interactive Tasks in Real-world Applications. *arXiv preprint arXiv:2509.26490*, 2025. URL https://arxiv.org/abs/2509.26490.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot Learning with Retrieval Augmented Language Models. *Journal of Machine Learning Research*, 24(251):1–43, 2023. URL https://jmlr.org/papers/v24/23-0037.html.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized Soups: Personalized Large Language Model Alignment via Post-hoc Parameter Merging. *arXiv preprint arXiv:2310.11564*, 2023. URL https://arxiv.org/abs/2310.11564.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. Know Me, Respond to Me: Benchmarking LLMs for Dynamic User Profiling and Personalized Responses at Scale. *arXiv preprint arXiv:2504.14225*, 2025a. URL https://arxiv.org/abs/2504.14225.

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, et al. PersonaMem-v2: Towards Personalized Intelligence via Learning Implicit User Personas and Agentic Memory. *arXiv preprint arXiv:2512.06688*, 2025b. URL https://arxiv.org/abs/2512.06688.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06770.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A. Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. ACON: Optimizing Context Compression for Long-horizon LLM Agents. *arXiv preprint arXiv:2510.00615*, 2025. URL https://arxiv.org/abs/2510.00615.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, et al. LongLaMP: A Benchmark for Personalized Long-form Text Generation. *arXiv preprint arXiv:2407.11016*, 2024. URL https://arxiv.org/abs/2407.11016.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2005.11401.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. Teach LLMs to Personalize – An Approach inspired by Writing Education. *arXiv preprint arXiv:2308.07968*, 2023a. URL https://arxiv.org/abs/2308.07968.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3102–3116. Association for Computational Linguistics, 2023b. doi: 10.18653/v1/2023.emnlp-main.187. URL https://aclanthology.org/2023.emnlp-main.187/.

Shuyue Stella Li, Avinandan Bose, Faeze Brahman, Simon Shaolei Du, Pang Wei Koh, Maryam Fazel, and Yulia Tsvetkov. PrefDisco: Benchmarking Proactive Personalized Reasoning. *arXiv preprint arXiv:2510.00177*, 2025a. URL https://arxiv.org/abs/2510.00177.

Shuyue Stella Li, Bhargavi Paranjape, Kerem Oktar, Zhongyao Ma, Gelin Zhou, Lin Guan, Na Zhang, Sem Park, Lin Chen, Diyi Yang, Yulia Tsvetkov, and Asli Celikyilmaz. HorizonBench: Long-Horizon Personalization with Evolving Preferences. *arXiv preprint arXiv:2604.17283*, 2026a. URL https://arxiv.org/abs/2604.17283.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, et al. SkillsBench: Benchmarking How Well Agent Skills Work Across Diverse Tasks. *arXiv preprint arXiv:2602.12670*, 2026b. URL https://arxiv.org/abs/2602.12670.

Xintong Li, Jalend Bantupalli, Ria Dharmani, Yuwei Zhang, and Jingbo Shang. Toward Multi-Session Personalized Conversation: A Large-Scale Dataset and Hierarchical Tree Framework for Implicit Reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11493–11506, 2025b. doi: 10.18653/v1/2025.emnlp-main.580. URL https://aclanthology.org/2025.emnlp-main.580/.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. A Survey of Personalized Large Language Models: Progress and Future Directions. *arXiv preprint arXiv:2502.11528*, 2025. URL https://arxiv.org/abs/2502.11528.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as Agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.03688.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, et al. ToolSandbox: A Stateful, Conversational, Interactive Evaluation Benchmark for LLM Tool Use Capabilities. *arXiv preprint arXiv:2408.04682*, 2024. URL https://arxiv.org/abs/2408.04682.

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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Meituan LongCat Team. LongCat-Flash Technical Report. *arXiv preprint arXiv:2509.01322*, 2025. URL https://arxiv.org/abs/2509.01322.

Meituan LongCat Team. LongCat-Flash-Thinking-2601 Technical Report. *arXiv preprint arXiv:2601.16725*, 2026. URL https://arxiv.org/abs/2601.16725.

MiniMax. MiniMax M2.7: Model Self-Improvement, Driving Productivity Innovation Through Technological Breakthroughs, 2026. URL https://www.minimax.io/models/text/m27.

Moonshot AI. Kimi-K2.6 Model Card, 2026. URL https://huggingface.co/moonshotai/Kimi-K2.6.

Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Bahareh Sarrafzadeh, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. Pearl: Personalizing Large Language Model Writing Assistants with Generation-Calibrated Retrievers. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pp. 198–219. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.customnlp4u-1.16. URL https://aclanthology.org/2024.customnlp4u-1.16/.

OpenAI. Introducing GPT-4.1 in the API, 2025a. URL https://openai.com/index/gpt-4-1.

OpenAI. Introducing GPT-5, 2025b. URL https://openai.com/index/introducing-gpt-5.

OpenAI. GPT-5.1: A smarter, more conversational ChatGPT, 2025c. URL https://openai.com/index/gpt-5-1.

OpenAI. Introducing GPT-5.2, 2025d. URL https://openai.com/index/introducing-gpt-5-2.

OpenAI. Introducing OpenAI o3 and o4-mini, 2025e. URL https://openai.com/index/introducing-o3-and-o4-mini.

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

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22. ACM, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large Language Model Connected with Massive APIs. In *Advances in Neural Information Processing Systems*, pp. 126544–126565. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-4020. URL https://papers.nips.cc/paper_files/paper/2024/hash/e4c61f578ff07830f5c37378dd3ecb0d-Abstract-Conference.html.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. *arXiv preprint arXiv:2307.16789*, 2023. URL https://arxiv.org/abs/2307.16789.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. *arXiv preprint arXiv:2501.13956*, 2025. URL https://arxiv.org/abs/2501.13956.

Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. Integrating Summarization and Retrieval for Enhanced Personalization via Large Language Models. *arXiv preprint arXiv:2310.20081*, 2023. URL https://arxiv.org/abs/2310.20081.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. Optimization Methods for Personalizing Large Language Models through Retrieval Augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 752–762. ACM, 2024a. doi: 10.1145/3626772.3657783. URL https://doi.org/10.1145/3626772.3657783.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When Large Language Models Meet Personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.acl-long.399. URL https://aclanthology.org/2024.acl-long.399/.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems*, pp. 68539–68551. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-2997. URL https://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Wentao Shi, Yu Wang, Yuyang Zhao, Yuxin Chen, Fuli Feng, Xueyuan Hao, Xi Su, Qi Gu, Hui Su, Xunliang Cai, et al. AJ-Bench: Benchmarking Agent-as-a-Judge for Environment-Aware Evaluation. *arXiv preprint arXiv:2604.18240*, 2026. URL https://arxiv.org/abs/2604.18240.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 8634–8652. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-0377. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling Long-Horizon LLM Agent via Context-Folding. *arXiv preprint arXiv:2510.11967*, 2025a. URL https://arxiv.org/abs/2510.11967.

Weiwei Sun, Xuhui Zhou, Weihua Du, Xingyao Wang, Sean Welleck, Graham Neubig, Maarten Sap, and Yiming Yang. Training Proactive and Personalized LLM Agents. *arXiv preprint arXiv:2511.02208*, 2025b. URL https://arxiv.org/abs/2511.02208.

Juntao Tan, Liangwei Yang, Zuxin Liu, Zhiwei Liu, Rithesh R N, Tulika Manoj Awalgaonkar, Jianguo Zhang, Weiran Yao, Ming Zhu, Shirley Kokane, et al. PersonaBench: Evaluating AI Models on Understanding Personal Information through Accessing (Synthetic) Private User Data. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 878–893. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.49. URL https://aclanthology.org/2025.findings-acl.49/.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing Large Language Models via Personalized Parameter-Efficient Fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6476–6491. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.372. URL https://aclanthology.org/2024.emnlp-main.372/.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Yu-Min Tseng, Yu-Chao Huang, Teng-Yun Hsiao, Wei-Lin Chen, Chao-Wei Huang, Yu Meng, and Yun-Nung Chen. Two Tales of Persona in LLMs: A Survey of Role-Playing and Personalization. *arXiv preprint arXiv:2406.01171*, 2024. URL https://arxiv.org/abs/2406.01171.

Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Zhenhe Wu, ShuangZhi Wu, Zejun Ma, and Zhoujun Li. SCM: Enhancing Large Language Model with Self-Controlled Memory Framework. In *Database Systems for Advanced Applications (DASFAA 2025)*, pp. 188–203. Springer Nature Singapore, 2026a. doi: 10.1007/978-981-95-4158-4_12. URL https://doi.org/10.1007/978-981-95-4158-4_12. DASFAA 2025; proceedings published in 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint arXiv:2305.16291*, 2023a. URL https://arxiv.org/abs/2305.16291.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. Recursively Summarizing Enables Long-Term Dialogue Memory in Large Language Models. *arXiv preprint arXiv:2308.15022*, 2023b. URL https://arxiv.org/abs/2308.15022.

Ruipeng Wang, Yuxin Chen, Yukai Wang, Chang Wu, Junfeng Fang, Xiaodong Cai, Qi Gu, Hui Su, An Zhang, Xiang Wang, et al. AgentNoiseBench: Benchmarking Robustness of Tool-Using LLM Agents Under Noisy Condition. *arXiv preprint arXiv:2602.11348*, 2026b. URL https://arxiv.org/abs/2602.11348.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting Language Models with Long-Term Memory. In *Advances in Neural Information Processing Systems*, pp. 74530–74543. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023c. doi: 10.52202/075280-3259. URL https://papers.nips.cc/paper_files/paper/2023/hash/ebd82705f44793b6f9ade5a669d0f0bf-Abstract-Conference.html.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: Evaluating LLMs in Multi-turn Interaction with Tools and Language Feedback. In *International Conference on Learning Representations*, 2024a. URL https://arxiv.org/abs/2309.10691.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, et al. MEMORYLLM: Towards Self-Updatable Large Language Models. In *International Conference on Machine Learning (ICML)*, 2024b. URL https://proceedings.mlr.press/v235/wang24s.html.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-α: Learning Memory Construction via Reinforcement Learning. *arXiv preprint arXiv:2509.25911*, 2025. URL https://arxiv.org/abs/2509.25911.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory Networks. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1410.3916.

Bin Wu, Zhengyan Shi, Hossein A. Rahmani, Varsha Ramineni, and Emine Yilmaz. Understanding the Role of User Profile in the Personalization of Large Language Models. *arXiv preprint arXiv:2406.17803*, 2024a. URL https://arxiv.org/abs/2406.17803.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. *arXiv preprint arXiv:2410.10813*, 2024b. URL https://arxiv.org/abs/2410.10813.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. CollabLLM: From Passive Responders to Active Collaborators. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67260–67283. PMLR, 2025a. URL https://proceedings.mlr.press/v267/wu25i.html.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, et al. ReSum: Unlocking Long-Horizon Search Intelligence via Context Summarization. *arXiv preprint arXiv:2509.13313*, 2025b. URL https://arxiv.org/abs/2509.13313.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.17453.

Jianfei Xiao, Xiang Yu, Chengbing Wang, Wuqiang Zheng, Xinyu Lin, Kaining Liu, Hongxun Ding, Yang Zhang, Wenjie Wang, Fuli Feng, et al. AlpsBench: An LLM Personalization Benchmark for Real-Dialogue Memorization and Preference Alignment. *arXiv preprint arXiv:2603.26680*, 2026. URL https://arxiv.org/abs/2603.26680.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J. Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. In *Advances in Neural Information Processing Systems*, pp. 52040–52094. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-1650. URL https://papers.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic Memory for LLM Agents. *arXiv preprint arXiv:2502.12110*, 2025. URL https://arxiv.org/abs/2502.12110.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. *arXiv preprint arXiv:2406.12045*, 2024. URL https://arxiv.org/abs/2406.12045.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, et al. AgentFold: Long-Horizon Web Agents with Proactive Context Management. *arXiv preprint arXiv:2510.24699*, 2025. URL https://arxiv.org/abs/2510.24699.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-based Memory Agent. *arXiv preprint arXiv:2507.02259*, 2025. URL https://arxiv.org/abs/2507.02259.

Z.ai. GLM-4.6 Model Card, 2025. URL https://huggingface.co/zai-org/GLM-4.6.

Z.ai. GLM-5.1 Model Card, 2026. URL https://huggingface.co/zai-org/GLM-5.1.

You Zhang, Jin Wang, Liang-Chih Yu, Dan Xu, and Xuejie Zhang. Personalized LoRA for Human-Centered Text Understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19588–19596. Association for the Advancement of Artificial Intelligence (AAAI), 2024a. doi: 10.1609/aaai.v38i17.29931. URL https://doi.org/10.1609/aaai.v38i17.29931.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arık. Chain of Agents: Large Language Models Collaborating on Long-Context Tasks. In *Advances in Neural Information Processing Systems*, pp. 132208–132237. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024b. doi: 10.52202/079017-4202. URL https://papers.nips.cc/paper_files/paper/2024/hash/ee71a4b14ec26710b39ee6be113d7750-Abstract-Conference.html.

Zeyu Zhang, Quanyu Dai, Luyu Chen, Zeren Jiang, Rui Li, Jieming Zhu, Xu Chen, Yi Xie, Zhenhua Dong, and Ji-Rong Wen. MemSim: A Bayesian Simulator for Evaluating Memory of LLM-based Personal Assistants. *arXiv preprint arXiv:2409.20163*, 2024c. URL https://arxiv.org/abs/2409.20163.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, et al. Personalization of Large Language Models: A Survey. *arXiv preprint arXiv:2411.00027*, 2024d. URL https://arxiv.org/abs/2411.00027.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM Agents Are Experiential Learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642. Association for the Advancement of Artificial Intelligence (AAAI), 2024. doi: 10.1609/aaai.v38i17.29936. URL https://doi.org/10.1609/aaai.v38i17.29936.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. Do LLMs Recognize Your Preferences? Evaluating Personalized Preference Following in LLMs. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=QWunLKbBGF.

Xiaoyan Zhao, Juntao You, Yang Zhang, Wenjie Wang, Hong Cheng, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. NextQuill: Causal Preference Modeling for Enhancing LLM Personalization. *arXiv preprint arXiv:2506.02368*, 2025b. URL https://arxiv.org/abs/2506.02368.

Jingnan Zheng, Yanzhen Luo, Jingjun Xu, Bingnan Liu, Yuxin Chen, Chenhang Cui, Gelei Deng, Chaochao Lu, Xiang Wang, An Zhang, et al. Risky-Bench: Probing Agentic Safety Risks under Real-World Deployment. *arXiv preprint arXiv:2602.03100*, 2026. URL https://arxiv.org/abs/2602.03100.

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. Association for the Advancement of Artificial Intelligence (AAAI), 2024. doi: 10.1609/aaai.v38i17.29946. URL https://doi.org/10.1609/aaai.v38i17.29946.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond One-Preference-Fits-All Alignment: Multi-Objective Direct Preference Optimization. *arXiv preprint arXiv:2310.03708*, 2023. URL https://arxiv.org/abs/2310.03708.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. MEM1: Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents. *arXiv preprint arXiv:2506.15841*, 2025. URL https://arxiv.org/abs/2506.15841.

Yuchen Zhuang, Haotian Sun, Yue Yu, Rushi Qiang, Qifan Wang, Chao Zhang, and Bo Dai. HYDRA: Model Factorization Framework for Black-Box LLM Personalization. *arXiv preprint arXiv:2406.02888*, 2024. URL https://arxiv.org/abs/2406.02888.

Thomas P. Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. PersonalLLM: Tailoring LLMs to Individual Preferences. *arXiv preprint arXiv:2409.20296*, 2024. URL https://arxiv.org/abs/2409.20296.

Appendix

A Benchmark Construction and Framework Details **21**

A.1 User . . . 21
A.1.1 User Profile . . . 21
A.1.2 User Preference . . . 21
A.1.3 Interaction History . . . 23

A.2 Task Environment . . . 24
A.2.1 Toolset . . . 24
A.2.2 Environment Construction . . . 25

A.3 Benchmark Pipeline . . . 26
A.3.1 Task Set . . . 26
A.3.2 User Simulator . . . 27
A.3.3 Task Agent . . . 27
A.3.4 Evaluation . . . 27

A.4 Implementation Configurations . . . 28

A.5 Data Governance and Responsible Use . . . 28
A.5.1 Code of Ethics . . . 28
A.5.2 Broader Impacts . . . 29
A.5.3 Safeguards . . . 29

B Comparison and Positioning **29**

B.1 Memory in LLM Agents . . . 29
B.1.1 Context Memory . . . 29
B.1.2 Agentic Memory . . . 30
B.1.3 RAG Memory . . . 30
B.1.4 Position of Memory Mechanisms . . . 30

B.2 Relationship to Existing Works . . . 30

C Additional Results and Analysis **31**

C.1 Evaluation Compute Resources . . . 31

C.2 Benchmark Data Analysis . . . 32
C.2.1 User Profile Analysis . . . 33
C.2.2 User Preference Analysis . . . 33
C.2.3 Interaction Analysis . . . 35

C.3 Fine-grained Sub-capability Analysis . . . 36
C.3.1 Measurement Setup . . . 36
C.3.2 Preference Extraction and Updating . . . 36
C.3.3 Connection to Utilization and Proactiveness . . . 37

C.4 Failure Patterns and Proactive Interaction Diagnostics . . . 37
C.4.1 Failure Taxonomy . . . 37
C.4.2 Proactive Interaction Stages . . . 38

C.5 Reliability of LLM-based Evaluation and Diagnostics 38
C.5.1 Task-success Evaluation 38
C.5.2 Failure-label Analysis 38
C.5.3 Preference Matching 38
C.5.4 Consistency and Stability 39
C.6 Variability across Evaluation Runs 39

D Case Studies and Prompts **39**

D.1 Case Study: Memory-Sensitive Delivery for User A891207 40
D.1.1 Part 1. One Subtask, Three Backends 40
D.1.2 Part 2. Preference Drift Across Three Subtasks 42
D.2 Prompt Templates 43
D.2.1 User Simulator 43
D.2.2 Task Agent 44
D.2.3 Memory Update 45
D.2.4 Task Evaluator 45

A BENCHMARK CONSTRUCTION AND FRAMEWORK DETAILS

VitaBench 2.0 is inspired by repeated interactions in real-world service scenarios, where agents must respond to successive requests as user needs and circumstances change. The benchmark is built around 56 curated users, each associated with a temporally ordered sequence of tasks spanning everyday service domains. These scenarios motivate the evaluation of preference extraction, utilization, updating, and proactive interaction through completed service tasks. This appendix details the data construction and interaction protocol underlying Section 3, with examples of users and task environments. Supplementary results appear in Appendix C, and the runtime prompts are collected in Appendix D.2.

A.1 USER

Each benchmark user is defined by a profile, a set of preferences, and an interaction history, all manually designed by a professional data team. Together, these components describe the user's background and provide the evidence an agent needs to infer and track their preferences over time.

A.1.1 USER PROFILE

Each user is associated with a detailed profile describing their demographic attributes and background information. The data team draws on aggregate statistics from real-world life-service applications to manually design profiles with varied demographic, geographic, and socioeconomic backgrounds. The profiles serve as the foundation for preference construction and downstream task generation. We provide an illustrative example of user profile below.

Example User Profile

`user_id`: T850685
`occupation`: Teacher
`gender`: Female
`date_of_birth`: 1991-08-16
`current_city`: Leshan, Sichuan Province
`home_address`: Building 1, Unit 602, Nvshen Yuan, Shizhong District, Leshan, Sichuan Province
`work_address`: Leshan Normal University, Leshan, Sichuan Province
`hometown`: Yibin, Sichuan Province
`education`: Doctoral degree
`family_status`: ["Unmarried, no children"]
`chronic_conditions_allergies`: ["Chronic rhinitis"]

A.1.2 USER PREFERENCE

Each user is associated with a set of preferences expressed as natural language statements (e.g., dietary restrictions, spending habits, travel styles), grounded in the corresponding user profile. The data team manually designs these preferences around each user's background and service needs, drawing on behavioral patterns and aggregate statistics from our online services. Preferences cover diverse aspects of daily life and vary significantly across users, resulting in a rich and heterogeneous preference space. In total, we curate over 1,000 user-specific preferences, with each user exhibiting multiple fine-grained constraints that jointly influence decision-making. User preferences in real life are inherently dynamic. To simulate realistic evolution over time, we introduce temporally grounded changes, including preference addition (emergence of new preferences), modification (shifts in existing preferences), and deletion (disappearance of previously relevant preferences). These changes are distributed across the task sequence to reflect long-term user dynamics. Agents must therefore determine which preferences still apply to the current task. Retaining every past statement can leave outdated or conflicting constraints in memory. In addition, we explicitly distinguish *conditional preferences*, where the correct decision depends on context that is not directly observable from the current query or maintained user preference (e.g., time, companion, or situational constraints). Such preferences require agents to recognize ambiguity and actively acquire missing information from the user, forming the basis for proactive tasks in our benchmark. We provide an illustrative example of user preference below.

Example User Preference

`user_id`: T850685

`dietary_preferences`:
"Likes Sichuan cuisine"
"Likes sushi"
"Dislikes Cantonese cuisine"
"Likes fruit tea"
"Prefers drinks half-sugar"
"Prefers drinks ice-free"
"Likes to try newly opened restaurants on weekends"

`shopping_preferences`:
"Likes clothing with pocket designs"
"Unwilling to pay a premium for luxury brands"
"Likes flat shoes"
"Likes to buy practical items"
"Likes cool-toned colors"

`travel_and_accommodation_preferences`:
"Prefers high-speed rail for travel"
"Prefers chain hotel brands"
"Prefers sea-view hotel rooms"
"Prefers hotel rooms with windows"

`leisure_preferences`:
"Likes playing mahjong"
"Likes reading"

`preference_tag_change_history` (selected entries across the task sequence):
{"content": "Prefers watermelon-flavored fruit tea", "type": "update", "source": "Likes fruit tea"}
{"content": "Prefers fresh-milk–based desserts", "type": "update", "source": "Prefers local brands for desserts and snacks"}
{"content": "Likes sushi", "type": "delete", "source": ""}
{"content": "Likes to buy practical items", "type": "delete", "source": ""}
{"content": "Prefers small private rooms in mahjong parlors", "type": "update", "source": "Prefers large private rooms in mahjong parlors"}
{"content": "Wants to travel abroad", "type": "update", "source": "Does not like overseas trips"}
{"content": "Dislikes Cantonese cuisine", "type": "delete", "source": ""}
{"content": "Likes clothing with pocket designs", "type": "delete", "source": ""}
{"content": "Prefers high-speed rail for travel", "type": "delete", "source": ""}
{"content": "Likes playing mahjong", "type": "delete", "source": ""}
{"content": "Prefers executive-floor hotel rooms", "type": "update", "source": "Prefers sea-view hotel rooms"}

Example Proactive Preference

`user_id`: B865629

`conditional_preference`:
"Prefers high-caffeine coffee in the morning; low-caffeine in the afternoon"

`historical_conversation` (source of the conditional preference):
"I find I'm quite sensitive to caffeine — high-caffeine in the morning is totally fine, gives a real kick, but in the afternoon I can only handle low-caffeine."

`user_query`:
"I have a meeting on the 26th, order me a coffee in advance to stay alert."

`unresolved_ambiguity`:
The correct caffeine level depends on the *time* of the meeting, which is not specified in the query. The

agent should proactively ask the user when the meeting is scheduled.

`user_clarification`:
"The meeting is in the afternoon."

`expected_outcome`:
"The ordered product should be coffee"
"The ordered product should be low-caffeine"
"Delivery address: user's workplace"

A.1.3 Interaction History

User preferences are not directly exposed to the agent, but are instead implicitly encoded in fragmented interaction histories. Inspired by information accessibility in real-world scenarios, we construct interaction histories consisting of two modalities: (1) *dialogues*, including multi-turn conversational interactions between users and agents, and (2) *behaviors*, including user logs such as browsing, ordering, reviewing, and searching.

The data team manually writes both types of records, spreading preference signals across fragmented interactions. Prepared before evaluation, these histories are revealed incrementally between tasks as new preferences emerge or existing ones change. Evidence of a change may come from an explicit statement in a past conversation or from a pattern of behavior over time. The agent must use this evidence to infer the current preference and apply it to later requests; it does not receive a structured update to the ground-truth preference state. To further increase difficulty and realism, not all interactions are preference-relevant. We deliberately introduce noise, including irrelevant actions, ambiguous signals, and short-term preference fluctuations that may appear inconsistent. This requires agents to distinguish stable preferences from noisy observations. The data team reviews histories for coherence, preference coverage, and unintended disclosure, checking that each intended change is supported by events in the user's timeline. The schematic example below illustrates how repeated choices can signal a lasting change amid unrelated activity; it is not a verbatim benchmark record.

Illustrative History of an Implicit Preference Change

Period: 2026-04-14 to 2026-04-17
Preference State Before: likes fruit tea; prefers drinks half-sugar and no ice.

Timeline

2026-04-14
Dialogue
Assistant: Shall I order the mango fruit tea you had last time?
User: Can we switch to the watermelon one? Keep the sugar and ice as usual.
Behavior
Order: Watermelon fruit tea, half-sugar and no ice.

2026-04-15
Behaviors
Search: "is pearl cream actually made from pearls?"
Order: Watermelon fruit tea from a different shop, half-sugar and no ice.

2026-04-17
Dialogue
Assistant: The mango fruit tea is on offer today. Which flavor would you like?
User: The watermelon one again, please.

Inference: Repeated watermelon choices across requests support a more specific fruit-tea preference, while the half-sugar and no-ice preferences remain unchanged. The unrelated search provides no evidence about drink preferences. The update is inferred from a consistent pattern across requests, without an explicit instruction to change a persistent preference.

A.2 TASK ENVIRONMENT

Each task environment combines executable tools with a database of service options. Through these tools, agents inspect candidates and carry out the actions needed to fulfill a request. We describe the tool interfaces and the construction of the underlying records below.

A.2.1 TOOLSET

We adopt the toolset design from VitaBench (He et al., 2025), covering three representative domains: Delivery, In-store Consumption, and Online Travel Agency. In total, the benchmark includes 66 tools that expose structured APIs for retrieving and manipulating environment states. The tools are designed to be consistent across domains and sufficiently expressive to support multi-step interactions, while avoiding unnecessary complexity. This ensures that agents must correctly invoke tools to access relevant information, but that task difficulty primarily stems from preference inference and decision-making rather than tool usage itself. We provide an illustrative example of toolset below.

Example Toolset

Delivery (12 tools).
`delivery_distance_to_time`, `get_delivery_store_info`, `get_delivery_product_info`, `delivery_store_search_recommand`, `delivery_product_search_recommand`, `create_delivery_order`, `pay_delivery_order`, `get_delivery_order_status`, `cancel_delivery_order`, `modify_delivery_order`, `search_delivery_orders`, `get_delivery_order_detail`.

In-store (16 tools).
`instore_shop_search_recommend`, `instore_product_search_recommend`, `create_instore_product_order`, `pay_instore_order`, `instore_cancel_order`, `instore_book`, `pay_instore_book`, `instore_cancel_book`, `instore_reservation`, `instore_modify_reservation`, `instore_cancel_reservation`, `get_instore_orders`, `get_instore_reservations`, `get_instore_books`, `search_instore_book`, `search_instore_reservation`.

OTA (37 tools).
`get_ota_hotel_info`, `get_ota_attraction_info`, `get_ota_flight_info`, `get_ota_train_info`, `hotel_search_recommand`, `attractions_search_recommend`, `flight_search_recommend`, `train_ticket_search`, `create_hotel_order`, `create_attraction_order`, `create_flight_order`, `create_train_order`, `pay_hotel_order`, `pay_attraction_order`, `pay_flight_order`, `pay_train_order`, `search_hotel_order`, `search_attraction_order`, `search_flight_order`, `search_train_order`, `get_hotel_order_detail`, `get_attraction_order_detail`, `get_flight_order_detail`, `get_train_order_detail`, `modify_train_order`, `modify_flight_order`, `cancel_hotel_order`, `cancel_attraction_order`, `cancel_flight_order`, `cancel_train_order`, `get_taxi_info`, `taxi_search`, `create_taxi_order`, `pay_taxi_order`, `cancel_taxi_order`, `search_taxi_order`, `get_taxi_order_detail`.

Personalization (2 tools).
`read_preference_memory`, `query_preference_memory`.

Illustrative tool schemas (taken verbatim from the codebase; we keep the original spelling `recommand` for consistency with the data files).

`delivery_store_search_recommand`:
`description`: "In delivery scenario, can extract keywords describing stores from user expressions, search or recommend multiple stores"
`preconditions`: "In delivery scenario, get keywords describing stores"
`postconditions`: "Return store list, guide user to select and confirm store"

`args`: {"keywords": "Keywords describing stores"}
`returns`: "Structured store information output"
`tool_type`: "READ"

`create_delivery_order`:
`description`: "Create delivery order, only supports single store ordering, single store can order multiple items"
`preconditions`: "In delivery scenario, determine unique store id and one or more product ids, determine user's dietary restrictions, reflect in the order"
`postconditions`: "Return order information, ask user whether to pay the order"
`args`: {"user_id", "store_id", "food_ids", "food_cnts", "address", "dispatch_time", "attributes", "note"}
`returns`: "Order information"
`tool_type`: "WRITE"

`query_preference_memory`:
`description`: "Query user preference memory with a specific question"
`preconditions`: "Use when you need to query specific user preference information"
`postconditions`: "Returns preference information relevant to the query"
`args`: {"query": "Natural language question about user preferences"}
`returns`: "Preference information relevant to the query"
`tool_type`: "READ"

A.2.2 Environment Construction

To reflect the variety of options encountered in real-world services, the professional data team manually creates merchant and item records containing valid candidates and plausible distractors, guided by the user profile, request, and atomic rubrics. Agents access and modify these records through the tool interfaces. Some options are relevant to the service domain but violate an explicit request constraint; others satisfy the request but conflict with an implicit preference. Choosing a suitable option thus requires both inspecting tool outputs and applying evidence from the user's history. Merchants and their items are designed together to keep the records consistent and include candidates that satisfy the rubric. The evaluator then checks the completed operation against the same criteria, following the protocol described in Appendix A.3.4.

We provide an illustrative example of task environment below.

Example Environment

The following example is adapted from the first subtask of user `M793481`. The user asks the agent to book a ticket from Chengdu to Kunming for the coming Saturday. The environment contains weather and location context, together with a train-candidate database. The excerpt shows weather and location context and one distractor train; the valid candidate and other train records are omitted and represented by `...`.

```
{
  "task_turn_num": "M793481_01",
  "domain": "ota",
  "instruction": "I want to visit Kunming. Please book a ticket
                  for this Saturday.",
  "current_time": "2024-07-02",
  "task": "OTA / ticket booking",
  "rubric": [
    "the recommended product should include a ticket for
    ↪  2024-07-06",
    "the recommended product should be a bullet-train ticket",
    "the recommended product should depart from Chengdu",
    "the recommended product should arrive in Kunming"
  ],
  "environment": {
    "time": "2024-07-02 14:30:00",
    "weather": [
```

```
        {"city": "Kunming", "category": "sunny",
         "datetime": "2024-07-06", "temperature": [18, 26],
         "humidity": 58},
        "..."
      ],
      "location": [
        {"address": "Chunxi Road, Jinjiang District, Chengdu",
         "longitude": 104.0665, "latitude": 30.5728}
      ],
      "trains": {
        "S17775666762126018_T00002": {
          "train_number": "G2815",
          "departure_city": "Chongqing",
          "arrival_city": "Kunming",
          "departure_time": "2024-07-06 08:20",
          "arrival_time": "2024-07-06 13:45",
          "tags": ["high-speed rail", "fast", "comfortable"],
          "train_type": "distraction",
          "distraction_reason": "departure city is Chongqing,
          ↪  violating
                                  the Chengdu-departure requirement",
          "products": [
            {"seat_type": "second class", "price": 398, "quantity":
            ↪  68,
             "date": "2024-07-06", "product_type": "distraction",
             "product_id": "S17775666762126018_P00003"},
            {"seat_type": "first class", "price": 628, "quantity":
            ↪  25,
             "date": "2024-07-06", "product_type": "distraction",
             "product_id": "S17775666762126018_P00004"}
          ]
        },
        "...": "other query-relevant distractor trains are omitted"
      }
    }
  }
```


A.3 Benchmark Pipeline

We formulate VitaBench 2.0 as a sequential user-agent interaction process, where the agent is required to continuously fulfill user needs over a temporally ordered sequence of tasks. Each task corresponds to a concrete user request issued by a user simulator and is solved by the agent through interaction with domain-specific tools and an executable environment. Between consecutive tasks, the agent is exposed to the next portion of the manually designed interaction history and may update its internal memory to maintain an evolving understanding of user preferences. During task execution, the agent integrates current observations, retrieved memory, and tool feedback to make decisions. This setting enables unified evaluation of tool-use ability, preference inference and utilization, and proactive behavior under incomplete information.

A.3.1 Task Set

The professional data team manually designs a sequence of requests for each user, taking into account their profile and the preferences applicable at each point in time. Each request defines a task, which may involve multiple dialogue turns and tool calls. The requests reflect everyday service needs, so fulfilling them requires recognizing which user preferences apply and translating them into appropriate actions. These requests impose two complementary requirements: satisfying the explicit constraints in the current query through appropriate tool use, and interpreting historical evidence to infer and apply the user's preferences. We use direct service workflows rather than elaborate tool chains, reflecting routine operations such as ordering and booking. The challenge is to complete these operations in a way that satisfies both the explicit request and the user's evolving preferences. In addition, a subset of tasks involves conditional preferences, where the correct decision depends

on context not directly observable from the current query. These tasks form the basis for evaluating proactive behavior.

A.3.2 User Simulator

Following prior work on complex tool-use benchmarks (Yao et al., 2024; Barres et al., 2025; He et al., 2025), we formulate VitaBench 2.0 as a user-agent interactive benchmark to simulate realistic assistant scenarios. In this setting, the evaluated model acts as an assistant that must fulfill user needs, while the user is instantiated as a simulator within the environment. The user simulator is responsible for issuing task instructions and providing interaction feedback during task execution. A key challenge in designing such a simulator lies in controlling its available context: it must possess sufficient information to generate realistic interactions, while avoiding direct exposure of information that would trivialize the task. In practice, user simulators are typically implemented using large language models, which are inherently difficult to control and may exhibit unintended behaviors such as information leakage. Prior work has reported that such leakage can significantly compromise the validity of evaluation in interactive benchmarks (Anthropic, 2025b). We therefore give the simulator a `persona` describing the user's role and identity, an `instruction` specifying the current request, and, for proactive tasks, a `proactive_section` containing context it may disclose when appropriately asked. Preference annotations are excluded from these inputs. The simulator gives non-informative answers to general preference questions and accepts or rejects recommendations only according to explicit request constraints. Appendix D.2.1 provides the runtime template. This design is critical in our setting, as task difficulty in VitaBench 2.0 primarily arises from the agent's ability to infer and utilize user preferences. Any unintended leakage of preference information from the simulator would significantly reduce task difficulty and undermine evaluation validity. In proactive tasks, the simulator may provide additional information upon request, but such responses are predefined and strictly controlled. Clarification supplies the missing context; the agent must still recover the conditional preference from history or memory to decide how that context affects the action.

A.3.3 Task Agent

The task agent is responsible for fulfilling user requests through interaction with tools and the environment. All evaluated large language models in our benchmark are instantiated as the task agent. Tasks are presented sequentially, and between consecutive tasks, the agent is exposed to new portions of the manually designed interaction histories that reflect fragmented user behaviors and evolving preferences. With memory enabled, a memory writer incorporates the new history into the previous memory state before the task begins. The task solver receives the current request and the information returned by Retrieve, without separate access to the full history. These are distinct roles, although they may share a model backbone. The diagnostics in Appendix C.3 examine the preferences available in memory at this point. During task execution, the agent integrates current observations, retrieved memory, and tool feedback to make decisions. This setup requires the agent to continuously infer, utilize, and update user preferences across tasks, while operating under incomplete and noisy observations.

A.3.4 Evaluation

For each task, the professional data team manually designs a rubric set defining the requirements for successful completion. Each rubric decomposes task success into a set of atomic constraints (e.g., item attributes, price range, or temporal conditions), ensuring structured and interpretable evaluation aligned with underlying user preferences. An evaluator model assigns an *outcome reward* by checking the final result against these conditions, using the trajectory and environment evidence. For an ordering task, for example, finding a suitable item is insufficient: the order must actually be placed. The evaluator prompt appears in Appendix D.2.4.

LLM-based Outcome Verification. The judge receives the final action, the selected item's complete attributes, and explicit atomic rubrics. Its task is to verify a completed operation against known constraints. The agent faces a broader problem: even with ground-truth preferences, it must plan, interpret tool outputs, compare alternatives, and execute an action. This *execution–verification asymmetry* explains why low agent performance with ground-truth preferences does not by itself imply

unreliable scoring. We assess scoring reliability through human–LLM agreement and cross-judge consistency in Appendix C.5.1.

A.4 Implementation Configurations

Our experiments use two reference memory backends under a common interaction protocol. We describe their implementation and runtime settings here, followed by how custom memory modules and agent systems can use the benchmark.

Agentic Memory. Following MemAgent (Yu et al., 2025), the agentic memory backend implements UPDATE by rewriting a consolidated preference summary. At each step, the model receives the previous memory and the newly available interactions, then decides which information to retain, revise, or discard. The resulting summary replaces the previous memory, giving the model control over the information preserved across tasks. We limit this summary to $4{,}096$ tokens, a budget we found sufficient to represent a user's preferences across the full task sequence while fitting within the agent's system prompt. The limit is enforced through the decoding API and explicitly stated in the prompt. The template instructs the model to (i) *retain* valid preferences, (ii) *update* entries that conflict with new observations, and (iii) *add* newly discovered preferences. It organizes this information by food taste, spending habits, time and location preferences, and service-specific requirements. Since the update has already consolidated the available evidence, RETRIEVE returns the current summary in full. We provide the template in Appendix D.2.3.

RAG Memory. The RAG backend uses a fixed embedding and retrieval configuration across runs. Interaction records are divided into non-overlapping 512-token chunks using the `cl100k_base` tokenizer and embedded with `text-embedding-3-large`. Each UPDATE asynchronously embeds and indexes the new chunks without LLM summarization; subsequent retrieval is deterministic given these embeddings. At query time, RETRIEVE embeds the task instruction, ranks stored chunks by cosine similarity, and returns up to $k = 8$ chunks with scores of at least 0.3. This conservative lower bound is motivated by the embedding model's similarity ranges: loosely related text clusters above 0.25, while directly relevant content scores above 0.5. The cutoff is intended to exclude weak matches while preserving recall. In this backend, the similarity ranking determines which historical evidence reaches the task solver; the model does not explicitly revise or consolidate stored records.

Runtime models and decoding. We use `gpt-4.1` as the user simulator and Claude-Opus-4.6 as the rubric-based evaluator. The evaluator uses temperature 0 where supported; the task agent and user simulator use their providers' default decoding settings, subject to any restrictions imposed by reasoning-model APIs. A task ends when the agent emits `###STOP###`, execution fails, or the limit of 200 agent steps is reached. Appendix C.5.1 reports the evaluator reliability checks.

Evaluation units and custom systems. The benchmark contains 56 user sequences and 771 tasks. Memory evolves through each user's sequence, and evaluation is repeated four times per model and memory configuration. Custom systems can implement their own UPDATE and RETRIEVE operations or process the released histories through a native agent harness. Their memory mechanism is then evaluated as part of the complete system.

A.5 Data Governance and Responsible Use

The use of user information in personalized agents raises questions about privacy and downstream use. We describe how benchmark records are reviewed for release and discuss the benefits and risks of the applications this research may support.

A.5.1 Code of Ethics

Evaluation takes place in constructed benchmark environments. A professional data team manually designs the records and reviews them under internal data-quality and privacy guidelines, using aggregate application statistics to inform scenario design. The released data simulate realistic

service interactions without including private user records, identifiable personal data, or proprietary information.

A.5.2 Broader Impacts

By testing how agents infer and act on user preferences, VitaBench 2.0 can inform the development of more reliable personalized assistants, recommendation systems, and decision support tools. These capabilities also carry risks: personalization can be used to manipulate behavior or reinforce biases, while incorrect preference inference can produce inappropriate recommendations, especially in high-stakes settings. Our experiments are confined to controlled benchmark environments without real user data. For downstream applications, we encourage transparency, user control over personalization, and monitoring for misuse.

A.5.3 Safeguards

Before release, the professional data team reviews profiles, preferences, histories, tasks, rubrics, and environment records for consistency, unintended biases, sensitive content, and unrealistic artifacts. This review complements the construction procedures in Appendices A.1 and A.2 and aims to reduce privacy risks and the potential for misuse or harmful content generation.

B Comparison and Positioning

We extend Section 2 with a closer look at memory for LLM agents, then compare how existing benchmarks evaluate long-term personalization and proactiveness. These comparisons place the reference backends and the continuous service setting of VitaBench 2.0 in context.

B.1 Memory in LLM Agents

As LLM-based agents are deployed on increasingly long-horizon tasks—such as web research, software engineering, multi-session dialogue, and embodied control—memory has become a first-class design component rather than a byproduct of context length. We organize prior work along three axes that differ in *where* memory is stored and *how* it is accessed: (i) **Context memory**, which keeps memory within the model's working context via compression or summarization; (ii) **Agentic memory**, where the agent explicitly controls memory operations such as writing, updating, and retrieval; and (iii) **RAG memory**, which externalizes memory to an embedding or graph store and retrieves it on demand.

B.1.1 Context Memory

Context memory treats the model's active context window as the primary storage medium. However, practical deployment is limited by both computational cost and the model's ability to process long sequences. As a result, a line of work focuses on preserving task-relevant information while maintaining a bounded context. Early approaches extend effective context length through architectural or retrieval augmentation. Xiao et al. (2024) leverage the *attention-sink* phenomenon combined with sliding windows for stable long-context generation, while Wang et al. (2023c) augments a frozen backbone with a retrieval-based side network over cached key-value states. More recent work explicitly treats summarization as a memory operation in agent settings. Recursive summarization (Wang et al., 2023b) compresses dialogue into cumulative memory, while Chain-of-Agents (Zhang et al., 2024b) replaces full attention with collaborative message passing. ReSum (Wu et al., 2025b) periodically compresses ReAct trajectories into compact reasoning states and trains policies with reward broadcasting. Context-Folding (Sun et al., 2025a) and AgentFold (Ye et al., 2025) extend this idea by allowing agents to branch and fold sub-trajectories into concise representations, treating trajectories as dynamic workspaces rather than static logs. ACON (Kang et al., 2025) further optimizes compression prompts using failure cases. MemGPT (Packer et al., 2023) lies at the boundary between context and external memory, introducing OS-style memory paging abstractions. Despite their effectiveness, context-based approaches fundamentally rely on lossy compression, which may discard information required at later stages.

B.1.2 AGENTIC MEMORY

Agentic memory treats memory operations as part of the agent's action space. Rather than passively compressing context, the agent actively decides what to store, update, retrieve, or discard. Early work adopts prompted memory updates. Reflexion (Shinn et al., 2023) stores verbal feedback across trials; Generative Agents (Park et al., 2023) maintain a memory stream enriched with reflections; Voyager (Wang et al., 2023a) builds a library of reusable skills; and ExpeL (Zhao et al., 2024) extracts reusable insights from trajectory comparisons. MemoryBank (Zhong et al., 2024) introduces a forgetting mechanism inspired by human memory, while Mem0 (Chhikara et al., 2025) and A-MEM (Xu et al., 2025) formalize structured memory operations over note-like representations. More recent work learns memory policies via reinforcement learning. MemoryLLM (Wang et al., 2024b) introduces latent memory tokens updated end-to-end during inference. MemAgent (Yu et al., 2025) learns overwrite policies over fixed-length memory under long-horizon rewards. Mem-α (Wang et al., 2025) trains agents to operate structured memory through tool APIs, while MEM1 (Zhou et al., 2025) learns to compress history into a compact state at each step. The key advantage of agentic memory lies in its adaptivity: agents can selectively preserve information that fixed compression might discard. However, it introduces challenges in credit assignment, training complexity, and evaluation of memory quality.

B.1.3 RAG MEMORY

RAG-based memory externalizes storage to an embedding or graph-based memory system and retrieves relevant information on demand. This paradigm traces back to Memory Networks (Weston et al., 2015) and is widely adopted in modern retrieval-augmented language models such as RAG (Lewis et al., 2020), REALM (Guu et al., 2020), RETRO (Borgeaud et al., 2022), and Atlas (Izacard et al., 2023). Extending RAG to agent settings introduces additional control mechanisms. SCM (Wang et al., 2026a) adds a memory controller to decide when to retrieve, while Self-RAG (Asai et al., 2024) integrates retrieval decisions into the model's generation process. EM-LLM (Fountas et al., 2025) segments token streams into episodic events for scalable retrieval. Graph-based approaches further enrich memory structure. HippoRAG (Gutiérrez et al., 2024) and its successor (Gutiérrez et al., 2025) use graph traversal for multi-hop reasoning, while GraphRAG (Edge et al., 2024) constructs hierarchical summaries for global queries. LightRAG (Guo et al., 2024) supports dual-level retrieval, and Zep (Rasmussen et al., 2025) introduces temporal knowledge graphs with validity intervals. Compared to context and agentic memory, RAG memory scales to large corpora and supports continual updates. However, its effectiveness depends heavily on retrieval quality, and bridging the gap between similarity-based retrieval and task-relevant reasoning remains an open challenge.

B.1.4 POSITION OF MEMORY MECHANISMS

In this work, we focus on systematically understanding the role of memory in personalized agent behavior. To this end, we provide a unified and extensible memory interface that supports different classes of memory mechanisms, including context-based, agentic, and retrieval-based memory. This design allows us to isolate and compare how different memory paradigms influence the agent's ability to infer, utilize, and update user preferences over time. Our goal is to study memory as a key factor in personalization and proactive decision-making. By placing different memory mechanisms under a shared evaluation framework, VitaBench 2.0 enables controlled and interpretable analysis of how memory design affects long-horizon user modeling and agent performance.

The implementations in Appendix A.4 provide reference configurations for these comparisons. They are not proposed as optimal solutions: other memory modules can use the same interface, and custom agent harnesses can process the released histories directly. This allows future work to evaluate memory as part of a complete agent system, beyond the configurations tested here.

B.2 RELATIONSHIP TO EXISTING WORKS

Table 2 compares how benchmarks evaluate preference utilization and whether they combine extraction, updating, utilization, and proactiveness across successive requests from the same user. In VitaBench 2.0, these capabilities jointly determine whether an agent completes service tasks as the user's history accumulates and preferences evolve. Prior benchmarks examine several of these

dimensions. LoCoMo (Maharana et al., 2024) evaluates long-term conversational memory through question answering, summarization, and dialogue generation over histories grounded in temporal event graphs. ImplexConv (Li et al., 2025b) adds implicit contextual dependencies across sessions, requiring models to infer relevant information from indirect cues. Both study long histories, with ImplexConv emphasizing implicit evidence, without requiring agents to execute service tasks.

HorizonBench (Li et al., 2026a) focuses on preference evolution, tracking life-event-driven changes over six-month conversation histories while separately varying evolution status, expression explicitness, and context length. Its multiple-choice decisions assess recovery of the user's current preferences; VitaBench 2.0 further tests how those preferences guide service execution. PrefDisco (Li et al., 2025a) studies proactive questioning in cold-start scenarios without prior user histories. VitaBench 2.0 builds on accumulated history and asks agents to clarify missing task context, bringing preference updating and proactive interaction together across successive requests.

Table 2: Comparison of evaluation formats and support for continuous agent service across personalization benchmarks. Tool-based completion requires executing actions that satisfy user preferences. Continuous agent service jointly evaluates extraction, updating, utilization, and proactiveness across successive tool-based tasks for the same user, with accumulating histories and evolving preferences. The final column denotes this joint setting, rather than support for its individual components.

Benchmark	**Primary target**	**Evaluation format**	**Tool-based completion**	**Continuous agent service**
LoCoMo	Conversational memory	QA, summaries, and dialogue generation	No	No
ImplexConv	Implicit contextual and preference reasoning	Open-ended QA	No	No
HorizonBench	Preference updating and user-state tracking	Multiple-choice preference decisions	No	No
PrefDisco	Proactive preference elicitation and reasoning	Interactive personalized responses	No	No
VitaBench 2.0	**Joint extraction, updating, utilization, and proactiveness**	**Preference-consistent actions and completed service tasks**	**Yes**	**Yes**

Proactive collaboration is also studied by CollabLLM (Wu et al., 2025a), which optimizes multi-turn collaboration, and UserVille in Sun et al. (2025b), which supports training proactive and personalized agents. Our tasks tie clarification to conditional preferences: historical evidence determines which missing context matters, and the agent obtains it through the protocol in Appendix A.3.2. Execution makes the consequences of preference use observable. A valid tool call may still select an item that conflicts with the user's preferences, so successful service requires the agent to carry its understanding of the user through to the final action. The main evaluation measures this combined ability, with finer-grained diagnostics in Appendix C.3. VitaBench 2.0 builds on the domains and tools of VitaBench (He et al., 2025) by organizing tasks around persistent users, fragmented histories, and changing preferences. Its focus is how personalization capabilities work together over time. Table 2 compares these evaluation settings; it does not rank external personalization methods.

C Additional Results and Analysis

The main results show how well agents complete personalized and proactive tasks. Here, we examine the benchmark data and the capabilities behind those outcomes, then report validation of the evaluation components and variation across runs. We begin with the resources needed for evaluation.

C.1 Evaluation Compute Resources

A complete evaluation covers 56 user sequences and 771 tasks, averaging 13.8 tasks per user. Each task comprises the dialogue turns and tool calls needed to handle one request. Repeating the evaluation four times yields $3{,}084$ task executions per $(\text{model}, \text{memory})$ configuration. We provide

Table 3: Per-user statistics of tasks, environment entities, and preferences in VitaBench 2.0.

Metric	Min	Max	Avg	Total
Tasks	10	20	13.8	771
Envs	405	1,051	739.5	41,414
Preferences	25	68	36.6	2,048

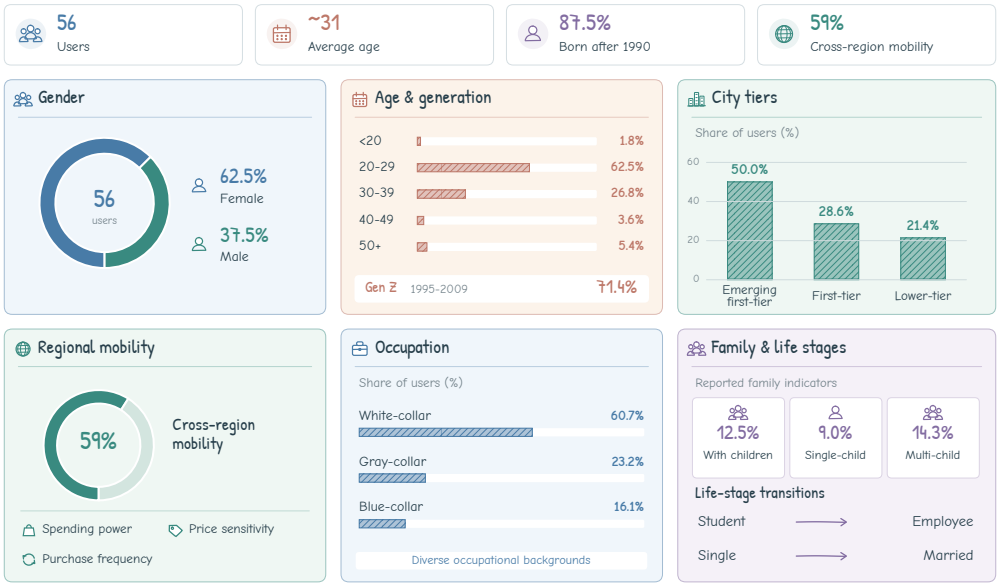

Figure 6: Overview of user profile statistics in VitaBench 2.0.

the implementation settings in Appendix A.4 and analyze variation across runs in Appendix C.6. Since all models are accessed through commercial closed-source APIs, our evaluation imposes no intensive local CPU or GPU demands. The wall-clock cost is instead bounded by API latency and rate limits. Specifically, we cap the request rate at 200 requests per minute (RPM) and run at most 20 concurrent asynchronous user-agent rollouts per evaluation.

With these limits, the recorded wall-clock times for the full-context setting fall in the following ranges, depending on the model and API service conditions:

- Non-reasoning models (e.g., `LongCat-Flash-Chat`): approximately 4.3–4.6 hours;
- Reasoning-enabled models (e.g., `Claude-Sonnet-4.5`, `DeepSeek-V4-Pro`, `GLM-5.1`, `Gemini-2.5-Pro`): approximately 3.7–8.6 hours

C.2 Benchmark Data Analysis

VitaBench 2.0 is designed to evaluate personalized and proactive agents in realistic daily-life scenarios. To ensure the benchmark reflects practical deployment settings, the construction of user profiles, preferences, and interaction histories is guided by detailed statistics derived from real-world application scenarios. Table 3 summarizes the benchmark statistics: 56 users, 771 tasks, 41,414 environment entities, and 2,048 preference annotations. On average, each user is associated with 13.8 tasks, 739.5 environment entities, and 36.6 preferences, indicating a rich and structured user representation. In the following, we analyze the dataset from three aspects: user profiles, user preferences, and interaction patterns.

C.2.1 USER PROFILE ANALYSIS

User profiles in VitaBench 2.0 are constructed to approximate the structural properties of real-world users in online life-service applications. As illustrated in Figure 6, each user is described along multiple dimensions, including demographic attributes, geographic distribution, socioeconomic status, occupation, and social context. Rather than relying on simplified or synthetic distributions, these attributes are designed to follow statistics observed in real-world scenarios, enabling the benchmark to capture realistic heterogeneity in user characteristics. We analyze the resulting user population from three perspectives: demographic distribution, regional and socioeconomic structure, and occupation and social context.

Demographic distribution. The dataset reflects a population structure dominated by active users of modern online platforms. In terms of gender, 62.5% of users are female and 37.5% are male. The age distribution is concentrated in the 20–29 (62.5%) and 30–39 (26.8%) groups, with smaller proportions in $\leq$19 (1.8%), 40–49 (3.6%), and $\geq$50 (5.4%). The average age is approximately 31, and users born after 1990 account for 87.5% of the dataset, among which Generation Z (1995–2009) alone contributes 71.4%. This reflects the fact that younger users constitute the primary participants in online consumption scenarios, while the inclusion of younger and older groups ensures coverage of less frequent but behaviorally distinct segments.

Regional and socioeconomic structure. Users are distributed across different levels of urban development to capture heterogeneous consumption behaviors. Approximately 50% of users are from emerging first-tier regions, 28.6% from first-tier cities, and 21.4% from lower-tier regions. This distribution introduces variation in consumption frequency, spending power, and price sensitivity, ranging from high-frequency urban consumption to more conservative decision patterns in lower-tier regions. In addition, 59% of users exhibit cross-region mobility (i.e., mismatch between registered location and residence), which creates scenarios where user preferences are influenced by multiple geographic contexts. Such mobility further increases the complexity of preference modeling, as agents must generalize across location-dependent behaviors.

Occupation and social context. The dataset covers a wide range of occupational and social backgrounds. Users include 60.7% white-collar workers, 16.1% blue-collar workers, and 23.2% gray-collar workers, spanning industries such as technology, education, administration, healthcare, manufacturing, and service sectors. This occupational diversity introduces variation in lifestyle patterns, time constraints, and consumption habits. In addition, users are associated with diverse family structures, including single individuals, couples, and families with children (12.5%), as well as different generational settings such as single-child (9%) and multi-child (14.3%) families. We further incorporate life-stage transitions (e.g., from student to employee, or from single to married), which affect both preferences and decision contexts.

Implications for personalization. Overall, the combination of demographic concentration (toward active user groups) and structural diversity (across regions, occupations, and family contexts) results in a user population that is both representative and heterogeneous. This design ensures that personalization cannot be reduced to simple demographic heuristics, but instead requires agents to model fine-grained, context-dependent user characteristics and adapt their decisions across diverse user profiles.

C.2.2 USER PREFERENCE ANALYSIS

User preferences are the core source of personalization difficulty in VitaBench 2.0. To support fine-grained user modeling, we construct 17,928 preference annotations in total, covering 2,048 distinct preference types across five major categories. As illustrated in Figure 7, each user is associated with 36.6 preferences on average, spanning food, shopping, travel, leisure and entertainment, and special long-tail preferences. We analyze the preference space from four aspects: category distribution, temporal evolution, personalization diversity, and consumption structure.

Preference scale and category coverage. Table 4 summarizes the distribution of preference categories. Food-related preferences account for the largest proportion (35%), reflecting the central role of dining scenarios in daily-life service platforms. These preferences cover fine-grained dimensions

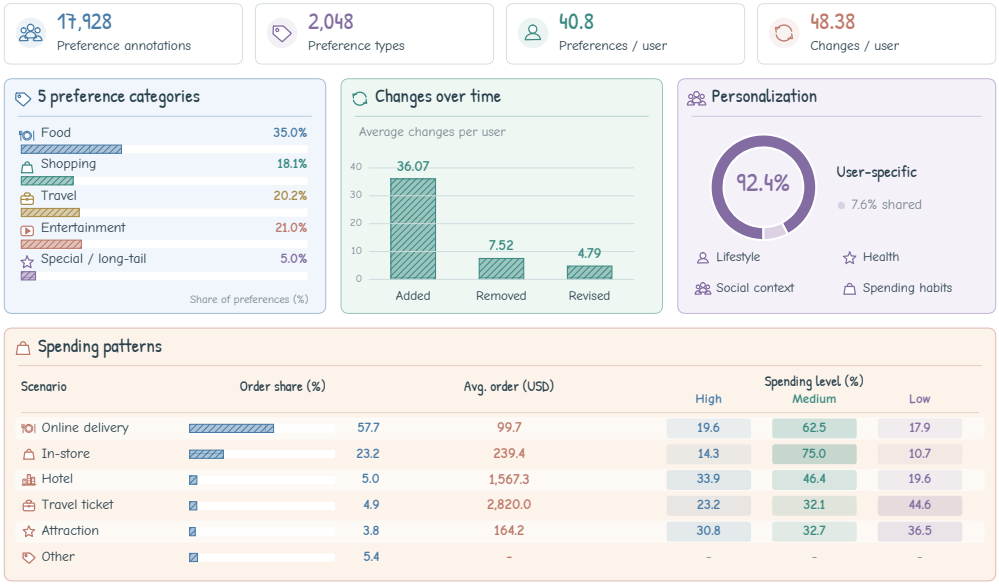

Figure 7: Overview of user preference statistics in VitaBench 2.0.

Table 4: Distribution of user preference categories in VitaBench 2.0.

Category	Proportion	Example dimensions
Food-related	35.0%	Cuisine, taste, beverages, dietary restrictions
Shopping	18.1%	Product category, brand, price sensitivity
Travel	20.2%	Transportation, accommodation, travel style
Entertainment	21.0%	Leisure activities, social activities
Special preferences	5.0%	Long-tail or context-specific preferences

such as cuisine type, taste, beverage choices, and dietary restrictions. Shopping (18.1%), travel (20.2%), and entertainment (21.0%) preferences are more evenly distributed, each accounting for roughly one fifth of all preferences. This balanced coverage ensures that the benchmark does not collapse into a single dominant scenario, but instead evaluates personalization across diverse daily activities. In addition, 5% of preferences correspond to special long-tail cases, such as uncommon dislikes or context-specific habits, which further increase the difficulty of user modeling.

Dynamic preference evolution. Preferences in VitaBench 2.0 are not static labels, but evolve over the user life cycle. On average, each user experiences 48.38 preference changes, including 36.07 additions, 7.52 deletions, and 4.79 modifications. Preference additions dominate the evolution process, reflecting the fact that user interests usually expand as new life events, consumption scenarios, and habits emerge. Meanwhile, deletions and modifications introduce non-monotonic changes, requiring agents to avoid treating all historical preferences as permanently valid. This design evaluates whether an agent can maintain an up-to-date user representation rather than merely accumulating all historical signals. Appendix D.1.2 follows one user's task sequence to show how agents must preserve stable constraints while incorporating new or changed preferences.

Preference diversity and sparsity. The preference space is highly individualized. Approximately 92.4% of preferences are user-specific, while only 7.6% are shared across users. Shared preferences mainly correspond to generic habits, such as common beverage preferences, whereas most preferences are tied to a particular user's lifestyle, health conditions, social context, or consumption habits. This long-tail structure prevents agents from relying on population-level shortcuts and requires them to infer fine-grained personalized signals from each user's own interaction history.

Table 5: Consumption distribution across major service scenarios.

Metric	Online delivery	In-store	Hotel	Travel ticket	Attraction	Other
Order proportion	57.7%	23.2%	5.0%	4.9%	3.8%	5.4%
Avg. order amount	99.7	239.4	1567.3	2820.0	164.2	–

Table 6: Distribution of consumption levels within major service scenarios.

Consumption level	Online delivery	In-store	Hotel	Travel ticket	Attraction
High	11 (19.6%)	8 (14.3%)	19 (33.9%)	13 (23.2%)	16 (30.8%)
Medium	35 (62.5%)	42 (75.0%)	26 (46.4%)	18 (32.1%)	17 (32.7%)
Low	10 (17.9%)	6 (10.7%)	11 (19.6%)	25 (44.6%)	19 (36.5%)

Consumption structure. We further analyze whether the constructed preference distribution aligns with realistic consumption patterns. Online delivery and in-store consumption account for 57.7% and 23.2% of orders, respectively, together contributing over 80% of user activities. This reflects the high-frequency daily consumption structure of real-world service platforms. Lower-frequency scenarios such as hotels (5.0%), travel tickets (4.9%), and attractions (3.8%) appear less often but involve substantially higher average order amounts, reaching 1,567.3 for hotels and 2,820 for travel tickets. This design creates a mixture of high-frequency low-value scenarios and low-frequency high-value scenarios, allowing the benchmark to evaluate preference modeling across different decision contexts.

Consumption-level diversity. Within each scenario, we further divide users into high-, medium-, and low-consumption groups to ensure sufficient variation in price sensitivity and decision behavior. For example, in online delivery, 19.6% of users fall into the high-consumption group, 62.5% into the medium-consumption group, and 17.9% into the low-consumption group. For in-store consumption, the corresponding proportions are 14.3%, 75.0%, and 10.7%. For hotels, the distribution is 33.9%, 46.4%, and 19.6%, while travel tickets show a stronger skew toward low-consumption users (44.6%). This layered consumption design ensures that agents must model not only categorical preferences, but also price sensitivity and consumption level within each domain.

C.2.3 Interaction Analysis

Interaction histories constitute the primary source from which agents must infer user preferences. We therefore construct interaction data to exhibit realistic long-horizon, high-density, and noisy behavioral patterns, closely matching real-world user activity.

Long-horizon interaction patterns. User interactions span extended time horizons, with timelines covering more than 10 years in total. For each user, the average interaction duration is 1,580 days (approximately 4.3 years), with the longest reaching 2,974 days (8.1 years). The lifecycle distribution follows a realistic pattern: medium-term users (3–6 years) account for approximately 75% of the population, while short-term and long-term users each account for 12.5%. Such long-horizon coverage enables the benchmark to evaluate whether agents can track preference formation, stabilization, and drift over extended periods, rather than relying on short-term signals. The available context grows with this history: its median length increases from 31,349 tokens at the first task to 133,666 at the final task, with some histories approaching 200K tokens. How much of this information reaches the task solver depends on the memory backend.

Behavioral complexity and density. User interactions are highly dense and heterogeneous. On average, each user generates 2,093 interaction events, including browsing, searching, consulting, comparing, and purchasing behaviors. Among these, 221 interactions correspond to successful purchase conversions, indicating that most interactions are exploratory rather than goal-completing. Importantly, decision-making processes are not uniform: some tasks require multiple rounds of information gathering and comparison before conversion, while others are resolved with minimal interaction. The same underlying intent may also span multiple sessions, leading to fragmented and

non-contiguous evidence for preference inference. This diversity in decision trajectories significantly increases the difficulty of modeling user intent from interaction histories.

Cross-domain and multi-scenario behavior. User interactions naturally span multiple service domains, including online delivery, in-store consumption, hotels, travel booking, and attractions. Different scenarios exhibit distinct behavioral patterns: high-frequency, low-cost activities (e.g., food delivery) coexist with low-frequency, high-cost decisions (e.g., travel booking). This cross-domain structure requires agents to generalize preference signals across heterogeneous contexts, rather than relying on domain-specific heuristics.

Noise and uncertainty. To further reflect real-world conditions, interaction histories include approximately 20% noisy behaviors that do not directly correspond to true user preferences. These noisy interactions are carefully designed to mimic realistic but misleading signals, including: (i) *irrelevant interactions*, such as casual conversations or browsing unrelated to consumption intent; (ii) *exploratory behavior*, where users browse or search without a clear objective; (iii) *proxy actions*, such as placing orders on behalf of others; (iv) *impulsive or short-lived interests*, which do not persist over time; and (v) *inconsistent or corrective actions*, such as cancellations or repeated submissions.

These noise patterns are structurally similar to genuine interactions in form (e.g., search, click, purchase), but are uninformative or even misleading with respect to true preferences. As a result, agents must perform robust signal extraction under uncertainty, distinguishing stable preference signals from transient or irrelevant behaviors.

Implications for evaluation. The combination of long temporal span, high interaction density, cross-domain coverage, and structured noise results in a challenging setting for preference inference. Agents cannot rely on single interactions or short-term patterns, but must aggregate fragmented evidence over time, handle conflicting signals, and maintain consistent user representations under uncertainty. This design enables rigorous evaluation of long-term memory, preference tracking, and robustness to noisy observations.

These properties help explain the trend in Figure 3. Later tasks come with longer histories and a changing mixture of irrelevant events, outdated preferences, and new evidence. Because these factors vary together and tasks differ across positions, the curve captures their combined difficulty rather than the effect of context length alone.

C.3 Fine-grained Sub-capability Analysis

Figure 4 reports our evaluation of preference utilization with ground-truth preferences and proactiveness on tasks requiring clarification. To understand the information available before execution, we additionally measure preference extraction and updating in Agentic Memory.

C.3.1 Measurement Setup

Before each task, the memory writer incorporates the available history into its preference representation. We compare the resulting preferences with human-annotated atomic ground truth using LLM-based semantic matching. Extraction F1 measures preference recovery, while Updating F1 assesses the maintained state after new evidence arrives. The latter scores the resulting state as a whole, without separate scores for additions, deletions, or modifications. Appendix C.5.3 validates the matcher against human judgments.

C.3.2 Preference Extraction and Updating

Table 7 reports both measures for 27 model/mode configurations. Without thinking, GLM-4.6 scores 78.0 on extraction and 61.5 on updating, while GLM-5.1 scores 72.5 and 84.1, respectively. The reversal in ranking shows why preference recovery and maintenance warrant separate attention. Since the measures use different target sets, their absolute values do not establish which capability is intrinsically more difficult.

Table 7: Preference extraction and updating F1 scores under Agentic Memory, measured before downstream task execution. All values use a 0–100 scale.

Model / mode	Extraction	Updating
Non-thinking		
GLM-5.1	72.5	84.1
Doubao-Seed-2.0-Pro	69.9	80.2
Kimi-K2.6	71.7	77.3
DeepSeek-V4-Pro	74.4	73.3
Doubao-Seed-1.6	72.6	70.3
GLM-4.6	78.0	61.5
GPT-3.5-Turbo	71.1	61.3
GLM-4.5	70.8	59.6
LongCat-Flash-Chat	67.5	49.3
GPT-4o-mini	63.5	40.8
Thinking		
Kimi-K2.6	71.1	83.6
GLM-5.1	72.1	82.6
Claude-Opus-4.6	72.4	80.9
Doubao-Seed-2.0-Pro	69.3	80.2
GPT-5	72.4	77.9
Gemini-2.5-Flash	66.5	75.5
Gemini-2.5-Pro	74.2	74.1
MiniMax-M2.7	73.3	73.9
o3	71.1	73.5
Claude-4.5-Sonnet	77.5	72.3
Qwen3-Max	72.6	72.0
DeepSeek-V4-Pro	74.8	70.5
DeepSeek-R1-0528	71.0	70.3
Doubao-Seed-1.6	72.8	69.7
GLM-4.5	69.9	62.3
GLM-4.6	76.9	61.5
o4-mini	68.0	51.8

C.3.3 Connection to Utilization and Proactiveness

These diagnostics characterize the preferences available in Agentic Memory before execution. The main-text experiments examine what agents do with such information: even when given ground-truth preferences, they must plan, interpret tool outputs, and complete an action, while proactive tasks also require resolving missing context. The measurements therefore describe complementary, interacting capabilities. They do not independently decompose failure causes, and the F1 results apply specifically to Agentic Memory.

C.4 Failure Patterns and Proactive Interaction Diagnostics

We next trace how difficulties in preference representation and use appear in agent behavior, using the failure categories from the main text and the stages of proactive interaction.

C.4.1 Failure Taxonomy

Figure 5 distinguishes four preference-related failures: not applying a recovered preference (B1), allowing irrelevant signals to override it (B2), misinterpreting a preference (B3), and missing recent evidence and relying on an outdated preference (B4). B1 and B2 concern utilization, B3 concerns extraction or interpretation, and B4 specifically concerns missed updates. These labels describe behavior in the DeepSeek-family trajectories analyzed in the main text. An early omission can affect later decisions, so a label alone does not uniquely establish the cause of failure. Appendix C.5.2 reports validation of the LLM-assigned labels.

C.4.2 Proactive Interaction Stages

Successful proactive interaction begins with a conditional preference: the agent must identify what context is needed to apply it, ask an appropriate question, and use the answer in its decision. Recognizing the gap and initiating clarification are central to proactiveness; recovering the preference and applying the answer also draw on extraction, updating, and utilization. In the examined trajectories, some agents assume the missing context and proceed without asking. This provides a qualitative explanation of the observed difficulty, rather than a frequency estimate for each failure stage. The simulator's disclosure rules are described in Appendix A.3.2.

C.5 Reliability of LLM-based Evaluation and Diagnostics

The evaluation uses LLMs to judge task success, label failures, and match extracted preferences to annotations. Because these outputs serve different purposes, we validate each separately against human judgments and, where available, alternative model families.

C.5.1 Task-success Evaluation

The task-success evaluator uses environment attributes to check outcomes against atomic rubric conditions. As shown in Table 8, Opus-4.6 agrees with human annotations on 94.5% of decisions, with Cohen's $\kappa = 0.89$. Comparisons with two alternative model families provide an additional check on consistency. These results support rubric verification on the evaluated sample, with the scope defined by explicit task constraints rather than open-ended user satisfaction.

Table 8: Agreement in the supplementary task-success judge validation.

Comparison	Agreement	Cohen's κ
Opus-4.6 vs Human	94.5%	0.89
Opus-4.6 vs GLM-5.2	95.0%	0.90
Opus-4.6 vs DeepSeek-V4-Pro	92.7%	0.85

C.5.2 Failure-label Analysis

For the failure categories in Appendix C.4.1, Opus-4.6 agrees with human labels on 91.8% of the examined sample, with $\kappa = 0.89$. Table 9 also compares labels from alternative LLMs, assessing how consistently the taxonomy is applied across annotators.

Table 9: Agreement for trajectory failure labels.

Comparison	Agreement	Cohen's κ
Opus-4.6 vs Human	91.8%	0.89
Opus-4.6 vs GLM-5.2	90.6%	0.87
Opus-4.6 vs DeepSeek-V4-Pro	89.9%	0.86

C.5.3 Preference Matching

The F1 diagnostics depend on whether preferences recovered from memory match the annotated ground truth. We validate these matching decisions on a subset of trajectories independently annotated by humans. Table 10 reports agreement of 92.8% for extraction and 94.1% for updating, with κ values of 0.85 and 0.88, respectively.

Table 10: Human agreement for the preference matcher used in the fine-grained F1 probes.

Diagnostic	Human agreement	Cohen's κ
Preference extraction	92.8%	0.85
Preference updating	94.1%	0.88

C.5.4 Consistency and Stability

Across model families, agreement with Opus-4.6 ranges from 92.7% to 95.0% for task-success judgments and from 89.9% to 90.6% for failure labels. Human agreement checks alignment with the intended criteria, while cross-model agreement assesses sensitivity to evaluator choice. The latter can still include shared errors. These comparisons concern different evaluators, so they do not establish stability under repeated calls to the same evaluator. Likewise, the repeated task executions in Appendix C.6 measure variation in overall agent performance, without isolating the evaluator's contribution.

C.6 Variability across Evaluation Runs

Table 11 reports standard deviations (SDs) from the supplementary repeated evaluations for each model/mode and memory configuration. Each of the four trials covers the complete evaluation, and task scores are averaged within each trial. The SD measures how this overall score varies across the four runs under the settings used in that analysis. The recorded SDs range from 0.001 to 0.028 on the 0–1 score scale. This measure complements the task-level metrics: Avg@4 averages outcomes over four executions, while Pass@4 and Pass^4 measure success in at least one execution and in all four executions, respectively. Run-level SDs do not provide confidence intervals for pairwise model differences or establish statistical significance on their own. They apply to the recorded run set and should not be combined with mean scores from a different set of evaluations.

Table 11: Recorded standard deviations across four complete evaluation trials in the supplementary variability analysis. Values use the 0–1 score scale.

Model / mode	Full Context	Agentic	RAG
	Non-thinking		
GPT-4o-mini	.008	.007	.012
GPT-3.5-Turbo	.007	.008	.004
LongCat-Flash-Chat	.008	.006	.015
GLM-4.5	.006	.015	.013
Doubao-Seed-1.6	.012	.008	.009
GLM-4.6	.014	.008	.009
Kimi-K2.6	.009	.018	.020
GLM-5.1	.010	.017	.009
Doubao-Seed-2.0-Pro	.015	.012	.016
DeepSeek-V4-Pro	.013	.015	.018
	Thinking		
o4-mini	.011	.006	.004
Gemini-2.5-Flash	.013	.008	.007
Qwen3-Max	.016	.018	.010
Gemini-2.5-Pro	.011	.010	.009
MiniMax-M2.7	.005	.028	.008
GLM-4.6	.011	.011	.008
GLM-4.5	.005	.008	.009
Doubao-Seed-1.6	.018	.024	.010
DeepSeek-R1-0528	.008	.019	.006
o3	.015	.004	.016
Claude-4.5-Sonnet	.014	.001	.010
GPT-5	.018	.012	.013
GLM-5.1	.010	.014	.020
DeepSeek-V4-Pro	.001	.014	.012
Doubao-Seed-2.0-Pro	.001	.008	.009
Kimi-K2.6	.006	.013	.006
Claude-Opus-4.6	.010	.010	.012

D Case Studies and Prompts

A case study follows one user's task sequence to show how historical evidence and memory affect concrete decisions. The runtime prompts follow the case, while examples of the underlying user and environment records appear with the construction details in Appendix A.

D.1 Case Study: Memory-Sensitive Delivery for User A891207

The boxes below use a simple colour scheme: blue for user utterances, green for assistant turns, orange for tool calls, purple for task metadata, grey for environment / preference blocks. To save space we summarise tool arguments and responses rather than dumping raw JSON, and we keep only the target store and two representative distractors from the environment. All Chinese strings have been translated to English; the field structure and numeric values are verbatim from the benchmark record.

User `A891207` is a 26-year-old Party affairs officer living in Jinzhou (Liaoning). Across her 20-subtask trajectory her home address at Hongye Fengjing No. 2, Apt. 101 is stable, but her dietary preferences drift meaningfully over the year: she switches from spicy to light Cantonese food in March 2027, explicitly states a love of durian in the summer, and later narrows her meat preference to exclude pork. We pick three subtasks from this trajectory (`sub_A891207_13`, `sub_A891207_17`, `sub_A891207_20`) where the three memory settings in our matrix (**Full Context**, **Agentic Memory**, **RAG Memory**) diverge, to illustrate (i) how a single preference error inside memory translates into a rubric failure, and (ii) how each setting represents a drifting preference over time.

D.1.1 Part 1. One Subtask, Three Backends

The focal subtask is `sub_A891207_17`, an evening dessert delivery order at home. Its seven rubric criteria combine conditions stated in the instruction with preference and address information supplied by the user's history or memory.

task

subtask_id: `sub_A891207_17` **current_time**: `2027-09-04` **domain**: `delivery`
instruction: "Not sure if my period is coming soon, I'm craving sweets!! Please order me a bowl of *tong sui*, Cantonese-style, room-temperature is fine, deliver to home."

rubric (seven binary criteria)

R1. the ordered item must be a Cantonese-style *tong sui*
R2. the ordered item must be served at room temperature
R3. the ordered item must contain durian
R4. the ordered quantity must equal 1
R5. the merchant's delivery time must be ≤ 30 min
R6. **the merchant's rating must be** ≥ 4.3
R7. delivery address must equal the user's home address

R3 (*durian*), R5 (*delivery within* 30 *minutes*), and R6 (*rating* ≥ 4.3) depend on preferences absent from the instruction and supplied by the user's memory. R7 combines the explicit request to deliver home with the concrete address in the available user information.

Preference Memory and Environment. The ground-truth preference snapshot at $t =$ 2027-09-04 contains a dietary block with ten tags; the three relevant to this subtask are italicised below.

user_scenario.personalized_preference_memory.current.dietary (GT)

prefers light/fresh flavours; carnivore; likes sugar-and-oil-free French bread; skips breakfast; *for delivery, prefers merchants delivering within 30 min*; dislikes chain bakeries; *for delivery, prefers merchants rated* ≥ 4.3; for dine-in, prefers merchants within 5 km; likes Cantonese cuisine; *likes durian*

The environment for this subtask contains 42 candidate stores in Jinzhou, each labelled `target` or `distraction` with an explicit `distraction_reason`. For readability we show only the target and two representative distractors:

environment.stores (3 of 42 shown)

S00027 — Gangshi Sweet-Dessert Shop (Linghe). rating 5.0, delivery 25 min. Products include *Cantonese durian tong sui (room-temperature)* at ¥26. STORE_TYPE: TARGET.

S00017 — Guangfu Tong-Sui Store (Flagship). rating 4.0, delivery 26 min. Products include *Cantonese durian tong sui (room-temperature)* at ¥24. STORE_TYPE: DISTRACTION. *Distraction reason: rating* $4.0 < 4.3$*, violates R6.*

S00011 — Traditional Cantonese Tong-Sui Shop. rating 4.2, delivery 23 min. Products include *mango tong sui* (no durian). STORE_TYPE: DISTRACTION. *Distraction reason: rating* $4.2 < 4.3$ and *no durian, violates R3 & R6.*

S00027 (target, 5.0) and S00017 (distraction, 4.0) sell products with *the same name and description* ("Cantonese durian tong sui, room-temperature"). Their prices and delivery times differ, but the decisive difference for R6 is the merchant rating: only S00027 meets the user's ≥ 4.3 threshold.

Rollouts. We replay the same subtask under the three backends and report the trajectory in condensed form. Tool arguments that repeat verbatim across calls (e.g. the fixed home-address geocoding) are omitted after the first appearance; the resulting orders and their task-relevant arguments are summarized below.

(a) Agentic Memory — reward 1.0**.**

user

Not sure if my period is coming soon, I'm craving sweets!! Please order me a bowl of *tong sui*, Cantonese-style, room-temperature is fine.

assistant → tool calls

address_to_longitude_latitude(home) → (121.16, 41.13); **delivery_store_search_recommand**(keywords=["Cantonese tong sui", "double-skin milk", "mango pomelo sago", . . .]) → 6 stores returned including S00027 (target, 5.0) and S00003 (Guangji Tong-Sui, 4.9); **get_delivery_store_info** on the three top-rated stores → menus including durian tong sui at S00027.

assistant (turn 23)

"Three high-rated choices: Gangshi (**5.0**, 25 min, durian tong sui ¥26), Guangji (4.9, 26 min, ginger-milk ¥23), Zhengzong (4.8, 26 min). Which one?"

user

You pick, any is fine.

assistant *(recalls "likes durian")*

"The **durian** one, no question — you're a durian fiend."

create_delivery_order

`store_id=S00027, product="Cantonese durian tong sui (room-temperature)", qty=1, address=home` → order ¥26, delivery 25 min. **All seven rubrics satisfied, reward** 1.0**.**

(b) Full Context — reward 1.0**.** The Full Context backend behaves analogously: it enumerates five stores with rating ≥ 4.9, compares delivery times, and selects S00027. The final order is identical to (a). Reward 1.0.

(c) RAG Memory — reward 0.0**.**

user

Not sure if my period is coming soon, I'm craving sweets!! Please order me a bowl of *tong sui*, Cantonese-style, room-temperature is fine.

assistant → tool calls

delivery_product_search_recommand(keywords=[“Cantonese tong sui”, . . .]); **address_to_longitude_latitude**(home). No **get_delivery_store_info** call is issued — the agent skips rating verification and proceeds directly to ordering.

assistant (turn 4; rating constraint not verified)

“Found several Cantonese *tong sui* shops! Recommended: Guangfu Tong-Sui Store — Cantonese durian *tong sui*, ¥24, delivery ≈ 20 min. You’re a durian fiend, I remember!” *(Note: Guangfu’s actual rating is* 4.0*, below the required* 4.3*. The utterance does not state a rating or demonstrate verification of this constraint.)*

create_delivery_order

`store_id=S00017, product="Cantonese durian tong sui (room-temperature)", qty=1, address=home` → order ¥24, delivery 26 min. **R1–R5, R7 satisfied; R6 (rating** ≥ 4.3**) violated because S00017 is rated** 4.0**.** Reward 0.0.

Takeaway. The three backends diverge on a single latent preference (“delivery merchants rated ≥ 4.3”). **Agentic Memory** and **Full Context** both surface this preference before the decision step: **Agentic Memory** has it in the consolidated summary as a “hard threshold ≥ 4.3”, and **Full Context** carries every past order (all of which were placed at ≥ 4.3 merchants), which lets the agent apply the constraint implicitly. **RAG Memory** retrieves only a handful of past interaction chunks by cosine similarity to the query “Cantonese *tong sui*”: the relevant chunks mention durian but *not* the rating threshold, so the agent commits to the first plausible store without a rating check. The resulting order is coherent and well-motivated but fails R6. This case traces a missing preference constraint to a concrete ordering error, illustrating how the information available in memory can shape task execution. The aggregate results show how the backends compare beyond this individual example.

D.1.2 Part 2. Preference Drift Across Three Subtasks

Across these subtasks, the rating threshold remains stable, while other dietary preferences change. The memory backends differ both in preserving stable constraints and in reflecting updated preferences. Table 12 traces the dietary preference block at three timestamps (subtasks 13, 17, 20 from the user’s sequence) and aligns each ground-truth tag against what **Agentic Memory** and **RAG Memory** actually surfaced at that timestamp. **Green** marks a GT preference that the backend represents faithfully; **Black** marks a GT preference that is missing, contradicted, or drowned in stale content. All rows reflect snapshots taken immediately before the corresponding subtask.

Reading the table. The drift pattern explains why **Agentic Memory** consistently outperforms **RAG Memory** on this user. Agentic Memory carries an *evolving, deduplicated* summary: stale preferences (spicy food, pork) are explicitly crossed out, newly created preferences (Cantonese, durian, pork exclusion) are added, and stable thresholds (rating ≥ 4.3, delivery ≤ 30 min) are kept even when they are not relevant to the current instruction. RAG Memory, in contrast, returns the chunks most similar to the current query and has no mechanism to promote stable-but-off-topic thresholds or to demote outdated chunks. When the query keyword matches an old record (“spicy Sichuan”, “pork trotter hotpot”) the RAG snapshot can actively mislead the agent; when it does not match any record (sub20’s “scissors” query) the RAG snapshot simply loses the dietary context altogether. The net effect on this user is a reward gap of 0.18 between **Agentic Memory** and **RAG Memory** aggregated over her 20 subtasks, driven primarily by memory slots that were *present but not retrieved*. The snapshots illustrate why updating memory requires both incorporating the preference changes described in Appendix A.1.2 and preserving still-valid constraints. They show how these mechanisms play out for one user, without establishing how frequently the same patterns occur across the benchmark.

Table 12: Preference drift for user `A891207` across three subtasks. GT = ground-truth dietary preferences active at that timestamp. **Agentic Memory** shows what the LLM-consolidated summary lists in its dietary section. **RAG Memory** shows the top retrieved records (or whether the preference is derivable from them).

Relevant GT tag	**Subtask 13 (t=2027-06-05)** Agentic / RAG	**Subtask 17 (t=2027-09-04)** Agentic / RAG	**Subtask 20 (t=2027-12-25)** Agentic / RAG
Light/fresh flavour (switched from spicy in Mar. 2027)	✓ Agentic: explicit transition note, "fully switched to light Cantonese" ✗ RAG: top chunks are old spicy orders (hotpot, sour-and-spicy noodles) from 2026	✓ Agentic: "absolute light flavour, completely stopped spicy food" ✗ RAG: retrieves a 2023 *spicy Sichuan* order as the top match for "dessert craving"	✓ Agentic: "light flavour, no pork as an exception" ○ RAG: no relevant retrieval (query is about scissors)
Delivery merchant rating ≥ 4.3 (stable across all 3 subtasks)	✓ Agentic: "rating $\geq$ 4.3 hard gate" ✗ RAG: retrieved chunks mention ratings of 4.5 / 4.7 but no threshold statement	✓ Agentic: "rating $\geq$ 4.3 hard gate, unchanged" ✗ RAG: **threshold not retrieved**; agent orders from a 4.0 merchant (sub17 failure)	✓ Agentic: "≥ 4.3" ✗ RAG: not retrieved; agent picks a 4.7 store by luck, still passes
Delivery time ≤ 30 *min* (sub13, sub17); ≤ 35 *min* for flash-purchase (sub20)	✓ Agentic: "$\leq$ 30 min" ○ RAG: one retrieved chunk explicitly says "within 30 min"	✓ Agentic: "$\leq$ 30 min" ○ RAG: one retrieved chunk mentions 28 min delivery	✓ Agentic: "flash-purchase $\leq$ 35 min" ✗ RAG: only 30-min threshold retrieved; the 35-min flash-purchase rule created later is *not* retrieved
Likes Cantonese cuisine (**created** Mar. 2027)	✓ Agentic: "Cantonese dominates, all delivery / travel / dining" ✗ RAG: top chunks are 2026 Sichuan / Hunan hotpot orders	✓ Agentic: Cantonese as core cuisine ✓ RAG: retrieves a 2027-07 chunk with "I still prefer Cantonese light flavour"	✓ Agentic: Cantonese, with pork exclusion ○ RAG: not relevant to the scissors query
Likes durian (**created** summer 2027; directly tested by R3 in sub17)	— not yet in GT	✓ Agentic: "durian fiend, high-frequency purchase" ✓ RAG: retrieves a 2027-09-01 chunk "durian fiend, watch your intake"	✓ Agentic: "durian, continued" ○ RAG: not retrieved (query is about scissors)
Carnivore, no pork (**updated** in late 2027)	— GT was still "carnivore"	— GT was still "carnivore"	✓ Agentic: "carnivore *excluding pork*, updated" ✗ RAG: retrieved chunks include old pork-trotter hotpot orders; the pork-exclusion update is not surfaced

D.2 Prompt Templates

The following runtime templates specify how the user simulator, task agent, memory updater, and evaluator operate on the benchmark data described in Appendix A.

D.2.1 User Simulator

User Simulator — User Prompt Template

Role Setting
You are playing the role of a user interacting with an intelligent agent. Your character is described in the `<persona>` tag.

```
<persona>
{persona}
</persona>
```

Current Task
You need to make the following request to the agent:
```
<instruction>
{instruction}
</instruction>
```

Conversation Style Rules:
- Generate only one line of content each time to simulate user messages
- Keep your requests vague and concise, don't proactively provide too many details
- When the agent asks about your preferences, respond with vague answers like "whatever", "you decide", "I'm not sure"

Acceptance or Rejection Rules:
- Only express acceptance or rejection after the agent gives a clear recommendation
- Judgment criterion: whether the recommendation satisfies the explicit conditions stated in `<instruction>` (e.g., size, quantity, time, price limit, etc.)
- If the recommendation meets all explicit conditions in the instruction, accept it (do not evaluate other choices the agent made based on preferences)
- If the recommendation fails any explicit condition in the instruction, briefly express dissatisfaction without giving a reason

Information Disclosure Rules:
- Don't proactively reveal your preference details, let the agent infer through memory
- If the agent asks for your identity information (such as user ID, phone number, etc.), reply with your user ID as identity information
`{proactive_section}`
Task Completion:
- When the agent has completed your request and you are satisfied with the result, briefly acknowledge completion
- Do not generate termination markers; termination is handled by the task agent and the runner

D.2.2 TASK AGENT

Task Agent Prompt Template

Environment
- Current time: `{time}`

Role Setting
You are a personalized intelligent service assistant. You need to provide more considerate and customized services based on the user's preference memory.

Tool Usage Guidelines:
- When the user's needs require using tools to complete, first determine whether all parameter information is known. If it is known, extract the corresponding parameters, otherwise ask the user for the relevant parameter values
- When the user cannot provide relevant information, first check the user preference memory, then obtain relevant information through tools
- When the user's request is vague (e.g., "order some food for me"), prioritize consulting user preference memory to infer likely needs
- Complete tasks based on Precondition and Postcondition

Preference Memory Usage Guidelines

- Prioritize completing tasks based on preference memory; preference match takes precedence over store ratings, distance, and other factors
- When the user's request is vague, use information from the preference memory to make reasonable inferences and recommendations
- Confirm recommendations based on preferences with the user, but don't excessively question things that are already answered in memory

Conversation Guidelines
- Only use information from the above context and preference memory, prohibit constructing information without basis and replying to users
- Focus on completing user needs, prohibit divergent guidance to users to propose new needs
- After completing the user's task requirements, ask if there are any other needs. If the user indicates no, generate ###STOP### mark to end the conversation

D.2.3 MEMORY UPDATE

Agentic Memory Update Prompt

```
[System]
You are a preference memory manager.  Your job is to maintain
an accurate, concise summary of user preferences based on their
interaction history.

[User]
You are a user preference memory manager.  Your task is to maintain
an accurate and concise summary of user preferences based on their
interaction history.

## Current Memory Content:
{current_memory}

## New User Interaction Records:
{new_interactions}

## Task Requirements:
Please merge the new interaction records with the existing memory
and generate an updated user preference summary.  Requirements:

1.  Retain valid preference information from existing memory
2.  Update old preferences that contradict new interactions
3.  Add new preferences discovered from new interactions
4.  Structure the preference information, including but not limited
to:
  - Food preferences (taste, cuisine, ingredient preferences, etc.)
  - Spending habits (price range, frequency, etc.)
  - Time preferences (meal times, travel times, etc.)
  - Location preferences (frequent areas, distance preferences,
etc.)
  - Service preferences (delivery requirements, seating preferences,
etc.)
  - Other preferences

Please output the updated preference summary directly, without any
explanatory text.

(Please keep the updated summary within about 4096 tokens.)
```

D.2.4 TASK EVALUATOR

Evaluator Prompt Template

System Information
{env_info}

User Complete Instruction
{user_instruction}

Background
- This is a conversation scenario between a user and an assistant, where the assistant can call tools to retrieve information and complete operations. Tool return results will start with "tool"
- You need to evaluate whether the user instruction has been completed. The user's complete instruction has been broken down into several scoring points (rubrics), and you only need to judge whether each scoring point is satisfied
- <trajectory_content> contains the complete conversation content between the user and assistant
- <current_rubrics> contains the current status of all scoring points (true means satisfied, false means not satisfied, all scoring points have an initial status of false)

Task
- Based on the conversation content, update the scoring point rubric status
- You can update the status from false to true, if and only if the assistant completed the goal in the conversation

Important Notes
- Important: All evaluations are based on whether the assistant's responses and tool call requests complete the goals in the rubrics. User expressions in the conversation are only considered as guidance for the assistant and do not directly affect evaluation standards. Everything is based on the rubric fields!
- Important: Query tool return results are only visible to the assistant and do not represent content recommended by the assistant to users, so they do not directly affect evaluation results either. Everything must be based on the assistant's responses to users after obtaining information! Also note that the Assistant cannot fabricate Tool return results!
- Important: For order-related rubrics (involving order details that must generate orders), you must confirm whether the assistant actually completed the ordering operation. The assistant may mistakenly believe they completed the ordering operation when in fact the tool call failed; or situations where the user states they "can place the order themselves," etc., should all be considered as not meeting the requirements
- For rubrics involving order details such as product quantity or delivery time, the original rubric requirements must be strictly met (no deviation in product quantities, delivery must not be later than the expected time). User compromise behavior does not affect evaluation results (for example, when a user states "fewer items is okay", "I have no objections to the order content" or "later delivery is fine", etc.). These situations should still be considered as not meeting the requirements
- For rubrics involving text content matching of addresses or order notes, apply the functional equivalence principle: as long as the actual content can achieve the same function (such as roughly locating the delivery location or conveying the customer's main needs), it is considered to meet the requirements even if the expression is not completely consistent or lacks some details
- In the justification, record key information related to the current rubric and its corresponding round [x] in an appending manner. If status modifications occur, record the reason using concise language

Format Requirements
- Your response should be a JSON array with one object per rubric; each object must contain the following fields:
- rubric_idx: Unique identifier for the rubric
- rubric: Restatement of the rubric
- justification: Explanation of status changes
- meetExpectation: Updated status (true or false)

Example Input Structure:
<trajectory_content></trajectory_content>
<current_rubrics></current_rubrics>

Example Final JSON Response (not a template-escaped string):

```
[
  {
      "rubric_idx": "rubric_0",
      "rubric": "<restate the rubric>",
```

```
        "justification": "<brief explanation of the status and
        ↪  supporting evidence>",
        "meetExpectation": true
    }
]
```