HABITUSMEM: BAYESIAN BELIEF MEMORY OVER EVOLVING USER PREFERENCES FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalized LLM agents must act on a user's current preferences, which evolve across interactions and are rarely stated in full. Existing memory systems store and retrieve content from the interaction history, leaving the agent to work out from fragments which preferences still hold and where they apply. Drawing on Bourdieu's account of habitus as durable but revisable dispositions, we model the user as a latent preference state that every interaction, including the current request, both reveals and updates. Bayesian inference over this state yields the assistant's belief about the user, and Bayesian decision theory specifies what memory should convey: the part of the belief that bears on the agent's current decision. We implement this view in HABITUSMEM, a supervised writer that renders the belief as text records for a frozen agent, each stating a claim with its evidence, scope, and status. The writer's training tasks mirror belief inference, belief update, and decision projection, and one writer serves both personalized question answering and personalized agent tasks. Experiments on PersonaMem-v2 and VitaBench 2.0 with two agent families show that HABITUSMEM outperforms existing memory systems in personalized question answering and yields the most reliable task completion for personalized agents among all non-privileged conditions, surpassing even full-history access.

1 INTRODUCTION

Large language model (LLM) agents increasingly assist the same user across many interactions, answering questions and carrying out tasks on the user's behalf (Jiang et al., 2025a;b; Chen et al., 2026). Useful assistance depends on the user's preferences and constraints, which a new request rarely states in full. These preferences are not a fixed list of facts. They form through repeated choices, persist across situations, and change over time: a user who has moved supplies a new delivery address, a user who has lost interest in a topic asks the assistant to stop raising it, and a gift chosen for a friend reveals little about the user's own taste. An agent must therefore act on the user's current preferences, which it can only infer from what the user has said and done.

Existing memory systems carry this information across interactions by storing content derived from the history: they extract facts, maintain notes or summaries, or retrieve and compress past passages for the current request (Zhong et al., 2024; Chhikara et al., 2025; Xu et al., 2026a; Lewis et al., 2020; Jiang et al., 2024). Such memory records what the user said or did, not what the history implies about the user now (Figure 1), and this leaves three problems unresolved. First, content does not convey how strongly the history supports a claim. A preference stated repeatedly and one inferred from a single purchase enter memory in the same form, so a gift bought for a friend can be recorded as the user's own taste. Second, content does not convey where a claim applies or whether it still holds. Update operations typically overwrite or delete an earlier entry, which conflates a preference that has been replaced with one that the user has only asked the assistant to stop using; otherwise, both entries remain and the agent faces conflicting statements. An instruction given for a single order is likewise indistinguishable from a standing preference. Third, content is written or selected without regard to the decision it will inform. A static summary can describe a user at length yet omit the preference on which a request depends: in PersonaMem-v2, a 1,935-character summary of one user lists occupation, education, and hobbies but omits the vegetable garden that the user's later request for relaxing activities at home calls for (Appendix C.3). Retrieval and compression select passages

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

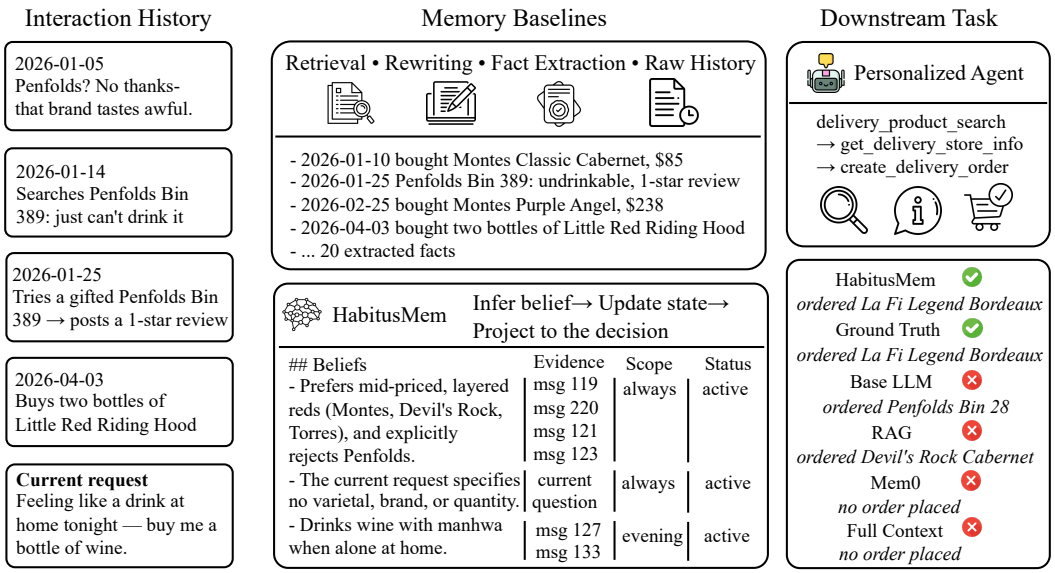


Figure 1: **Content-based memory versus belief memory on a personalized order.** The user has repeatedly rejected Penfolds wine and now asks for a bottle of wine without naming a brand. Content-based memories pass on fragments such as past purchases and a negative review, and the agents using them, like the base LLM, order the wrong wine or none at all. HABITUSMEM infers that the user prefers mid-priced layered reds and rejects Penfolds, records the evidence, scope, and status of each belief, and leads the agent to the same order as the ground-truth profile.

that match the request without resolving whether they are still current, and summaries can replace the exact product, variant, or quantity that a transaction requires with a paraphrase.

These problems have measurable consequences. Reading the full history does not resolve them: on PersonaMem-v2, frontier LLMs with long context windows reach only 37–48% accuracy on implicit personalization, and the bottleneck is reasoning about the user rather than context length (Jiang et al., 2025b). Content-based memory does not relieve the agent of this reasoning either. In our evaluation, each of five established memory systems is less accurate than full history on PersonaMem-v2 and achieves lower four-trial all-success than full history on VitaBench 2.0 (Section 4). These systems often reach a correct transaction in some trials but rarely in all of them: with the Qwen agent, Rewrite succeeds at least once on 61% of VitaBench subtasks but in all four trials on only 7%.

The root cause is that these designs represent the history rather than the user. Bourdieu's concept of *habitus* describes what personalization actually targets: durable dispositions, formed through past experience, that generate practices in concrete situations (Bourdieu, 2020). These dispositions are durable but not fixed: new experiences are perceived through categories formed by earlier ones and, in turn, reinforce or modify them (Bourdieu & Wacquant, 1992, p. 133). For an assistant, this suggests that the user has a latent preference state shaped by the whole interaction history, and that every interaction, including the current request, is a step in its evolution. This view is naturally captured by Bayesian modeling (Berger, 1987): earlier interactions form a prior belief, each new interaction updates it, and the posterior summarizes what the history supports about the user's present state. Memory should therefore convey this belief rather than the underlying content, and Bayesian decision theory specifies which part to convey: since the agent should minimize expected loss under the belief, memory should state what bears on its current decision. This view addresses all three problems. A claim's *evidence* conveys its credibility; its *scope* and *status* convey where it applies and whether later observations have left it active, superseded, or retracted; and the decision posed by the request determines which claims to include.

We instantiate this view in HABITUSMEM, a supervised writer that renders the belief as records of the form `claim | evidence | scope | status` for a frozen agent (Figure 2). Because exact inference over natural-language histories is intractable, the writer learns to approximate this Bayesian

target from four training tasks that mirror belief inference, belief update, and decision projection, and a single writer serves both personalized question answering and personalized agent tasks. On PersonaMem-v2, HABITUSMEM is the most accurate compact memory system under both agents, and on VitaBench 2.0 it achieves the most reliable task completion among non-privileged conditions, exceeding even full-history access. Our contributions are as follows:

- We propose to treat personalized memory as the assistant's belief about an evolving user state. Drawing on habitus, we formulate memory writing as Bayesian belief tracking followed by decision-theoretic projection, which specifies that each claim should carry its evidence, scope, and status.
- We introduce HABITUSMEM, a single supervised writer that renders this belief as structured records for a frozen agent and is trained on four tasks that mirror belief inference, belief update, and decision projection.
- We show on PersonaMem-v2 and VitaBench 2.0, with two agent families, that HABITUSMEM is the most accurate compact memory system for personalized question answering and yields the most reliable task completion for personalized agents.

2 RELATED WORK

2.1 PERSONALIZED LLM-POWERED AGENTS

Personalization has become a central axis along which LLM-based agents are extended into user-aligned decision systems, spanning profile modeling, memory, planning, and action execution (Xu et al., 2026b). Profile modeling constructs user representations either through persona-based descriptions of stable traits, as in FSPO (Singh et al., 2026), RLPA (Zhao et al., 2026b), and CURIO (Wan et al., 2026), or through response-based preference factorization over candidate outputs, as in RFM (Barreto et al., 2026), PReF (Shenfeld et al., 2025), and LoRe (Bose et al., 2025), and further defines the agent's own role as user-substituting or user-adaptive, exemplified by LD-Agent (Li et al., 2025a), PersonaAgent (Zhang et al., 2026), and PersonaX (Shi et al., 2025). Among these components, memory is widely regarded as the linchpin of personalization: because user preferences are revealed incrementally and evolve over time, an agent's ability to store, update, and retrieve user-relevant information across sessions directly determines whether downstream planning and execution remain aligned with the individual user rather than merely with the current turn. Building on this user- and memory-conditioned context, personalized planning either grounds decisions on existing profiles and memory in a single pass, as PRIME (Zhang et al., 2025), ALIGNXPLORE (Li et al., 2025b), and PbP (Xu et al., 2025a) do, or refines plans iteratively through clarification, user revision, or environment feedback, as in SAGE-Agent (Suri et al., 2026), Ask-before-Plan (Zhang et al., 2024), and AIPOM (Kim et al., 2025). Personalized action execution then instantiates these decisions through preference-aware tool selection and argument grounding, seen in PEToolLLaMA (Xu et al., 2025b), ToolSpectrum (Cheng et al., 2025), and AWARE-US (Kurmaz, 2026), along with outcome-level re-ranking or proactive suggestion, as in PEAR (Li et al., 2022) and ETAPP (Hao et al., 2025). Across these components, however, the reliability of planning and execution is bottlenecked by the memory that feeds them: a user-conditioned decision is only as good as the memory's account of how credible a stored preference is, in which situations it applies, and whether it still holds, distinctions that current personalization pipelines largely leave to ad hoc summarization or similarity-based retrieval rather than treating as first-class memory content.

2.2 AGENT MEMORY

A parallel line of work studies memory itself as a general capability of foundation agents, independent of a specific personalization objective. Existing systems are commonly organized along three dimensions (Huang et al., 2026). Memory substrate distinguishes external stores such as vector indices, text records, and structured or hierarchical stores from internal parametric memory realized as weights, latent states, or KV caches. Memory cognitive mechanism follows the sensory, working, episodic, semantic, and procedural decomposition inspired by human cognition. Memory subject separates user-centric memory, exemplified by MemoryBank (Zhong et al., 2024), A-MEM (Xu et al., 2026a), and Zep (Rasmussen et al., 2025), from agent-centric memory that accumulates the agent's own task experience, as in Voyager (Wang et al., 2023), ReasoningBank (Ouyang et al.,

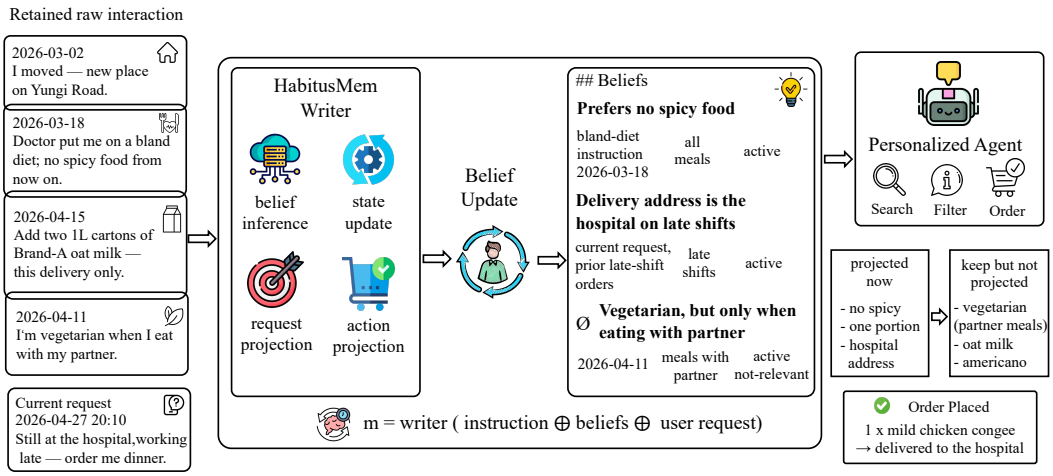


Figure 2: **Overview of HABITUSMEM.** A single writer, trained on belief inference, state update, request projection, and action projection, reads the retained raw interactions together with the current request and writes belief records with evidence, scope, and status. Only the beliefs that bear on the current request are projected to the frozen agent: the bland diet and the late-shift hospital address shape the dinner order, whereas the partner-specific vegetarian preference and the one-off oat-milk instruction are kept but not projected.

2026), and Agent Workflow Memory (Wang et al., 2024c). At the operational level, memory systems support storage and indexing, retrieval, update and refresh, compression and summarization, and forgetting, and recent work increasingly frames these operations as learned policies rather than fixed heuristics, whether through prompting as in MemGPT (Packer et al., 2023) and Mem0 (Chhikara et al., 2025), supervised fine-tuning that internalizes memory behavior into model parameters as in MemoryLLM (Wang et al., 2024b) and WISE (Wang et al., 2024a), or reinforcement learning that optimizes memory edits against downstream task or dialogue reward as in Memory-R1 (Yan et al., 2026), Mem-α (Wang et al., 2025), and MemAgent (Yu et al., 2026). Across this literature, however, memory is still predominantly treated as a content store: entries record what was said or observed without a principled account of how credible a claim is, where it applies, or whether later observations have superseded it, leaving content-based memory unable to distinguish a repeatedly stated preference from a one-off instruction or an active preference from one the user has since retracted.

3 METHODOLOGY

HABITUSMEM treats memory as the assistant's belief about the user's current preference state (Figure 2). Section 3.1 formalizes this belief and the decision it serves, Section 3.2 describes how a writer renders the belief as text records, and Section 3.3 describes how the writer learns the operations of belief tracking.

3.1 BELIEF TRACKING AND DECISION

Let z_t denote the user's preference state at interaction t, comprising the durable dispositions, current circumstances, and explicit constraints that shape what the user wants. The state evolves: a new experience may reinforce a disposition, modify it, or lead the user to withdraw it, which we describe by a transition $p(z_t \mid z_{t-1})$. The assistant observes the state only through interactions $o_t \sim p(o_t \mid z_t)$, such as statements, choices, corrections, and requests. The current request q_t is the most recent of these observations: it carries evidence about the user and also poses the decision that the agent must make. Given the history $H_t = (o_1, \ldots, o_t)$, the assistant's belief is the posterior

$$b_t(z_t) = p(z_t \mid H_t) \;\propto\; p(o_t \mid z_t) \int p(z_t \mid z_{t-1})\, b_{t-1}(z_{t-1})\, dz_{t-1}. \tag{1}$$

The recursion mirrors the account of habitus in Section 1: each observation is interpreted through the belief formed by earlier ones, and it either reinforces or revises that belief. The same posterior can be computed directly from the complete history, since exact batch and recursive inference yield identical beliefs.

The belief describes what the assistant knows, whereas the agent must act. A frozen agent receives the request, a memory m, and optional execution instructions g_τ, and produces an answer or action trajectory $a \sim A(\cdot \mid q_t, m, g_\tau)$, where τ indexes the task type. Let $\ell_\tau(a, z_t; q_t)$ measure how poorly an action serves a user in state z_t. Following Bayesian decision theory (Berger, 1987), the risk of a memory and the ideal memory are

$$\mathcal{R}_\tau(m; H_t) = \mathbb{E}_{z_t \sim b_t}\, \mathbb{E}_{a \sim A(\cdot \mid q_t, m, g_\tau)} \left[\ell_\tau(a, z_t; q_t)\right], \qquad m_t^\star \in \arg\min_{m \in \mathcal{M}} \mathcal{R}_\tau(m; H_t), \tag{2}$$

where $\mathcal{M}$ is the set of textual memories. Equation 2 separates two questions that content-based memory conflates. The belief determines how credible each claim about the user is, and the request determines, through the loss, which claims change the agent's decision. A gardening preference is equally credible whether the user asks for a weekend activity or a train ticket, but only the first request makes it relevant. Conversely, an order can fail on a single wrong variant or quantity, so the memory must convey such details exactly.

The formulation also determines what a textual memory should state about each claim. The agent needs the claim's *evidence*, the observations that support it, to distinguish a disposition established by repeated explicit statements from one inferred from a single choice. It needs the claim's *scope*, the situations in which the disposition guides behavior, because a disposition is expressed differently across situations and a choice made for one occasion should not steer every later request. It needs the claim's *status*, which records the evolution of the state: a claim may remain active, be superseded by a later preference, or be retracted when the user asks the assistant to stop using it, which restricts its use without making it false. These three fields define the record format of HABITUSMEM.

Exact inference over natural-language histories is intractable, and HABITUSMEM neither estimates numerical posteriors nor optimizes Equation 2 through agent rollouts. Equations 1–2 instead specify what the writer should produce, and supervised fine-tuning on demonstrated memories teaches it to produce such records (Appendix D).

3.2 WRITING THE BELIEF AS RECORDS

Memory construction starts from the accumulated raw interaction log. New observations are appended to the log, and a position- and budget-based rule T_τ selects the view $X_t = T_\tau(H_t)$ that fits the writer's input: on PersonaMem-v2 it keeps the beginning of each history, and on VitaBench 2.0 it keeps the beginning together with the most recent interactions (Section 4.1). This rule determines which evidence the writer can see, and the writer itself decides what enters the memory.

The writer receives the selected history, a format instruction I, and a suffix $s_\tau(q_t)$ containing the current request. For question answering, the suffix asks for facts and preferences that help answer the request. For agent tasks, an action suffix asks for explicit choices and fulfillment constraints, such as products, variants, quantities, addresses, and times, and asks the writer to keep explicit historical choices rather than inferred alternatives.

The output is a list of records under a `## Beliefs` header, one per line in the form `claim | evidence: source | scope: context | status: state`. The claim states a fact, preference, or constraint, and the remaining fields carry its evidence, scope, and status as defined above. The fields separate an enduring disposition, a one-off instruction, and a withdrawn preference. In an illustrative memory, a record that the user tends a small vegetable garden cites two earlier messages as evidence, applies to leisure at home, and remains active. A record of an order for two 1 L cartons of a specific oat milk keeps the exact quantity and variant that the transaction needs, and its scope, *this delivery only*, prevents it from steering later requests. A record that the user prefers spicy dishes, with the status *retracted (message 61)*, tells the agent that the preference should no longer guide its restaurant choices.

We instantiate the writer with Qwen3.5-9B (Qwen Team, 2026) and one LoRA adapter (Hu et al., 2021) shared across tasks. The entire memory is generated in one greedy decoding pass:

$$\hat{m}_t = \mathrm{GreedyDecode}[p_\theta(\cdot \mid I, X_t, s_\tau(q_t))], \qquad \hat{a} \sim A(\cdot \mid q_t, \hat{m}_t, g_\tau). \tag{3}$$

The agent receives the records together with the current request. On PersonaMem-v2, g_τ is empty. On VitaBench 2.0, g_τ instructs the agent to apply relevant constraints, to make a choice when the user delegates one, and to complete the requested transaction, and every memory system in our VitaBench comparison receives the same guidance. The 800-token cap applies to the generated records. Whenever the history or the request changes, the writer recomputes the memory from the selected raw log, which corresponds to the batch computation of Equation 1; repeated reads with the same history length and request reuse the cached memory.

3.3 Learning the operations of belief tracking

The writer learns from four tasks that correspond to the operations in Equations 1–2. *Direct writing* maps a raw history and a request to belief records, teaching inference from evidence. Its targets combine teacher memories with the preference annotations and update labels of the training personas, placing the annotated record first. *State update* maps previous records and a new history chunk to revised records, teaching the recursive update of Equation 1, including superseding and retracting earlier claims. *Request projection* maps accumulated records and a request to the records relevant to that request, and *action projection* does the same for requests that require execution while preserving exact choices and fulfillment constraints; both teach the decision step of Equation 2. Training continues from a parent writer that was trained on direct examples with the same record format and request conditioning.

To construct update and projection examples, the parent writer processes each training persona's history in 5,000-token chunks; a data builder merges the extracted records with the preceding ones using line-level deduplication and ordering heuristics, capping the state at 2,400 tokens, and each transition yields a state-update example. From the final state, the teacher writes up to eight request projections and four action projections per persona, each limited to 800 tokens. All targets come from the official PersonaMem-v2 training split.

Target preparation checks that the required fields are present and repairs or filters malformed records. These checks are syntactic and do not verify the factual correctness of the contents. Let $\mathcal{D}_k$ denote the four datasets before repetition. With weights $w = (1, 1, 1, 3)$, the writer minimizes the teacher-forced token loss

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\sum_{k=1}^{4} w_k \sum_{(x,m^\dagger)\in\mathcal{D}_k} \sum_{j} \log p_\theta(m_j^\dagger \mid x, m_{<j}^\dagger), \tag{4}$$

where the weight of three on action projection is implemented by the threefold repetition. All examples update one adapter, so that inference, update, and projection share parameters. Section 4.1 gives the training hyperparameters, and Appendix A gives further details on target construction and parameter allocation.

4 Experiments

We compare belief memory with other memory systems on personalized answering (Section 4.2) and personalized agent tasks (Section 4.3), and examine how writer training, request conditioning, and the form of retraction records affect the results (Section 4.4).

4.1 Experimental setup

Personalized benchmarks. PersonaMem-v2 (Jiang et al., 2025b) tests personalized answering over long histories with evolving preferences and requests to forget. We use its English 32k setting, train the writer on the official training split (18,500 queries), and evaluate on the 5,000-query benchmark split with multiple-choice (MCQ) accuracy and a generation score from a Qwen3.5-9B judge. VitaBench 2.0 (Chen et al., 2026) tests personalized agents that must infer, apply, and update preferences hidden in users' past dialogues and behavior logs while completing delivery, in-store consumption, and online travel tasks. We run all 771 subtasks of its 56 users four times each, with Qwen3.5-9B simulating the user and scoring each trajectory against the subtask rubric, and report the average reward (Avg@4), success in at least one trial (Pass@4, any-success), and success in all four trials (Pass^4, all-success). On both benchmarks the agent is DeepSeek-V4-Flash or Qwen3.5-397B.

Table 1: **PersonaMem-v2 MCQ accuracy by task scenario on the 5,000-query benchmark split.** Full context and the base LLM are reference conditions, and bold and underlined values mark the best and second-best compact memory systems in each column.

Memory	Know-ledge	Pro-Email	Pro-Writing	Crea-tive	Trans-lation	Personal-Email	Trouble	Chat	Social	Overall
Qwen3.5-397B										
Base LLM	0.644	0.408	0.381	0.420	0.438	0.433	0.393	0.396	0.386	0.446
Full context	0.519	0.594	0.573	0.543	0.624	0.585	0.599	0.548	0.565	0.569
RAG	0.436	0.481	0.453	0.462	0.486	0.470	0.437	0.450	0.474	0.460
Rewrite	0.519	0.503	0.466	0.466	0.479	0.469	**0.509**	0.440	0.485	0.484
Mem0	0.490	0.449	0.488	0.471	0.533	0.453	0.475	0.438	0.415	0.470
A-MEM	0.433	0.494	0.457	**0.510**	0.503	**0.530**	0.497	**0.502**	0.515	0.489
MemListener	0.443	0.399	0.422	0.373	0.438	0.404	0.459	0.388	0.423	0.418
HABITUSMEM	**0.612**	**0.544**	**0.532**	0.503	**0.540**	0.506	0.475	0.480	**0.522**	**0.530**
DeepSeek-V4-Flash										
Base LLM	0.368	0.230	0.236	0.257	0.274	0.232	0.246	0.265	0.224	0.266
Full context	0.533	0.512	0.514	0.512	0.553	0.545	0.529	0.516	0.524	0.527
RAG	0.468	0.371	0.343	0.396	0.410	0.372	0.355	0.402	0.361	0.391
Rewrite	0.513	0.419	0.440	0.401	0.419	0.415	**0.453**	0.394	0.441	0.438
Mem0	0.450	0.412	0.387	0.379	0.402	0.439	0.433	0.373	0.415	0.412
A-MEM	0.501	0.367	0.363	0.416	0.397	0.388	0.413	0.386	0.388	0.408
MemListener	0.415	0.357	0.383	0.338	0.402	0.350	0.405	0.335	0.374	0.376
HABITUSMEM	**0.630**	**0.431**	**0.464**	**0.462**	**0.458**	**0.471**	0.407	**0.428**	**0.470**	**0.479**

Because the static base writer and the base writer prompted with the request score below the base LLM in generation (Section 4.4), we treat MCQ accuracy as the primary PersonaMem-v2 measure.

Baselines. On both benchmarks we compare with the base LLM, in which the agent receives no memory, and with full context, in which the agent reads the history directly. On VitaBench 2.0, a ground-truth profile built from annotated preferences serves as an additional privileged reference. The compact memory baselines are retrieval-augmented memory (RAG), Rewrite, which maintains an LLM-rewritten summary of the history, Mem0, A-MEM, and MemListener (Lewis et al., 2020; Chhikara et al., 2025; Xu et al., 2026a; Zhao et al., 2026a). On PersonaMem-v2, we additionally compare writers that differ in training and request conditioning (Section 4.4).

Implementation. The writer is Qwen3.5-9B with one LoRA adapter (rank 64, alpha 128, dropout 0.05), trained for one epoch with learning rate 2×10^{-5}, a cosine schedule, and a 12,288-token sequence cutoff on eight A100 GPUs. The same adapter serves both benchmarks and decodes greedily with an 800-token output cap. On PersonaMem-v2, the writer reads at most 8,500 tokens from the beginning of the history and receives the request suffix. On VitaBench 2.0, it reads at most 11,400 tokens, allotting 15% of the history budget to the beginning of the history and the rest to the most recent interactions, and receives the action suffix. Appendix A gives further details.

4.2 PERSONALIZED ANSWERING

HABITUSMEM is the most accurate compact memory system. On the 5,000 benchmark queries (Table 1), HABITUSMEM has the highest overall accuracy among compact memory systems under both agents: 0.530 with Qwen, ahead of A-MEM at 0.489, and 0.479 with DeepSeek, ahead of Rewrite at 0.438. With DeepSeek it leads the compact systems in eight of the nine scenarios and trails Rewrite only in Trouble (0.407 versus 0.453). With Qwen it leads in five scenarios and trails the best compact system by 0.007 to 0.034 in Creative, Personal-Email, Trouble, and Chat. Its margin is largest in the Knowledge scenario (0.612 and 0.630, compared with at most 0.519 and 0.513 for the other compact systems), where the Qwen base LLM also reaches 0.644. Full context remains more accurate than every compact system in all scenarios except Knowledge (baseline details in Appendix B).

Table 2: **Personalized agent tasks on VitaBench 2.0** (771 subtasks, four trials each). Bold and underlined values mark the best and second-best compact memory systems in each column.

Memory	Qwen3.5-397B agent			DeepSeek-V4-Flash agent		
	Avg@4	Pass@4	Pass^4	Avg@4	Pass@4	Pass^4
Base LLM	0.0895	0.2010	0.0182	0.0833	0.1946	0.0130
Full context	0.3382	0.5888	0.1297	0.3119	0.5409	0.1102
Ground-truth profile	0.3330	0.5888	0.1193	0.3839	0.6316	0.1569
Compact memory systems						
RAG	0.2857	0.5391	0.0864	0.2251	0.4705	0.0562
Rewrite	<u>0.3139</u>	**0.6109**	0.0700	0.3123	<u>0.5863</u>	0.0869
Mem0	0.2844	<u>0.5694</u>	0.0636	0.2643	0.5707	0.0480
A-MEM	0.2964	0.5486	<u>0.0999</u>	<u>0.3165</u>	**0.5875**	0.0843
MemListener	0.2202	0.4189	0.0687	0.2422	0.4293	<u>0.0934</u>
HABITUSMEM	**0.3379**	0.5642	**0.1336**	**0.3619**	0.5824	**0.1505**

Full history remains stronger on multiple-choice questions. With the DeepSeek agent and the same judge for all conditions (Section 4.4), HABITUSMEM reaches 0.4788 MCQ accuracy and a generation score of 0.3745, 21.60 MCQ points and 0.0835 above the base LLM. Full context leads HABITUSMEM by 4.34 MCQ points, whereas HABITUSMEM scores 0.0728 higher in generation. A memory of at most 800 tokens thus supports personalized generation at least as well as the full history, whereas multiple-choice questions whose decisive evidence is lost during input selection or writing still benefit from the full history.

4.3 PERSONALIZED AGENT TASKS

HABITUSMEM yields the most repeatable task success. Table 2 reports all memory systems under both agents. HABITUSMEM attains the highest all-success of any condition without the privileged profile: 0.1336 with Qwen and 0.1505 with DeepSeek, compared with 0.1297 and 0.1102 for full context and 0.0999 and 0.0934 for the strongest compact baselines, A-MEM and MemListener. It also has the highest average reward among compact memory systems under both agents, 0.3379 and 0.3619, which matches full context with Qwen (0.3382) and exceeds it with DeepSeek (0.3119). The ground-truth profile remains stronger on all three metrics with DeepSeek, whereas with Qwen HABITUSMEM exceeds it in average reward and all-success.

Gains concentrate in all-success rather than any-success. Any-success does not follow the same pattern. With Qwen, Rewrite reaches more subtasks at least once than HABITUSMEM (0.6109 versus 0.5642); with DeepSeek, the two are nearly equal (0.5863 versus 0.5824). Yet HABITUSMEM exceeds Rewrite in all-success by 6.36 points under each agent. Among the subtasks solved in at least one trial, HABITUSMEM solves 24% with Qwen and 26% with DeepSeek in all four trials, compared with 11% and 15% for Rewrite and 22% and 20% for full context. The pattern suggests that records stating exact choices and constraints help the agent repeat a successful transaction more than they help it reach additional ones.

4.4 ABLATION STUDY

We examine two parts of the design: Figure 3 compares writers that differ in training and request conditioning, and Figure 4 relates the form of retraction records to accuracy.

Training the writer to use the request matters more than prompting it. With the DeepSeek agent, HABITUSMEM exceeds the static base writer by 14.96 MCQ points and 0.1933 in generation score. Giving the same base writer the current request recovers little of this gap, and HABITUSMEM still leads by 13.36 points and 0.1635. Neither training nor the request helps much alone: training a static writer raises MCQ accuracy only from 0.3292 to 0.3346, and the unadapted writer reaches similar accuracy with the current request (0.3452) and with an unrelated request (0.3474), so prompting it with the request yields no content-specific MCQ benefit. A matched unrelated-request comparison is

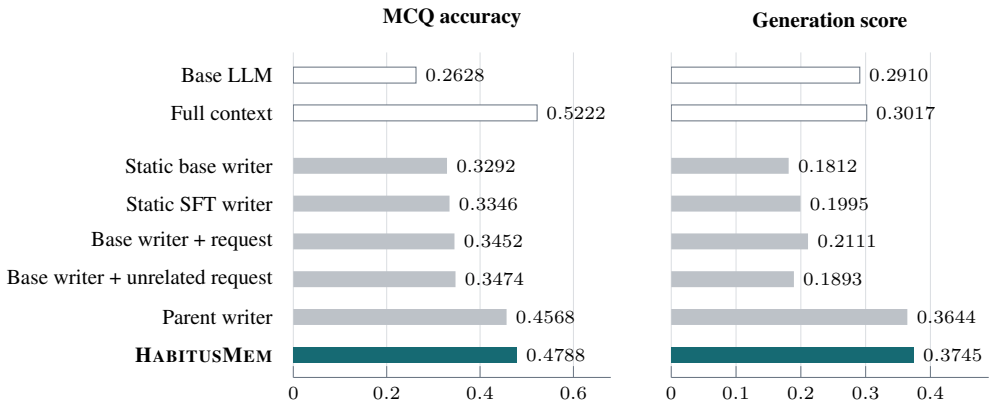


Figure 3: **Writer ablations on PersonaMem-v2** (5,000 benchmark queries, DeepSeek agent, same judge). Outlined bars are reference conditions, gray bars are writers that differ in whether they are trained and whether they see the current request, and the teal bar is HABITUSMEM. The writers compare successive training stages rather than isolated components.

not available for the trained writers. Most of the gain is already present in the parent writer, and the final four-task training adds 2.20 MCQ points and 0.0101 in generation score.

Retraction records must name what they revoke. Requests to forget remain a weakness of HABITUSMEM. On the PersonaMem-v2 forgetting queries with the Qwen agent, its MCQ accuracy (0.4914) trails that of A-MEM (0.6288) and full context (0.7090), although nearly every memory it writes contains a retraction record. Figure 4 indicates that the form of the record matters more than its presence. When the record names the revoked item, as the retracted spicy-food record in Section 3.2 does, accuracy reaches 0.7397; when it states only that some earlier preference was withdrawn, accuracy falls to 0.4532. The writer nevertheless produces the abstract form in 85.7% of its retraction records, although at least 73% of the training targets name the item, suggesting that it has learned to signal a retraction without learning to specify it. A second failure occurs when the memory still holds an active near-duplicate of the revoked preference: the agent then receives contradictory records, and accuracy drops to 0.2615. These comparisons are observational and do not separate errors of the writer from those of the agent. They nonetheless indicate that a status field is necessary but not sufficient, since the writer must also identify what it revokes and remove the records that the retraction supersedes.

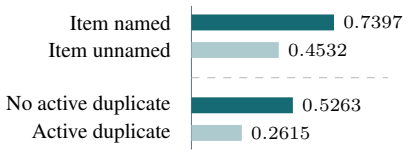


Figure 4: **Retraction form and MCQ accuracy** on PersonaMem-v2 forgetting queries (Qwen agent).

5 CONCLUSION

We introduced HABITUSMEM, which treats personalized memory as the assistant's belief about an evolving user state. Following the account of habitus as durable but revisable dispositions, it regards every interaction, including the current request, as evidence that reinforces or revises this belief, and a supervised writer states the part of the belief that bears on the agent's decision as records with evidence, scope, and status. On PersonaMem-v2, training this writer improves substantially over static and prompted writers. On VitaBench 2.0, it yields the most repeatable task success among the evaluated memory systems under two agent families. Two problems remain open: a record that marks information as retracted does not yet ensure that the agent stops using it, and input selection limits what any memory can contain.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs, to generate synthetic data sets, namely the supervised memory targets used to train the writer (Section 3.3), and to provide feedback on research methodology and experiments. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, implement methods, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, or interpret results. Additionally, we used generative AI tools to edit the writing of the paper to improve readability. We have reviewed all AI-assisted work. Specifically, all AI-suggested proofs were independently re-derived and verified for correctness by the authors before inclusion; AI-generated feedback on the experimental methodology was critically evaluated and incorporated only where the authors judged it sound; and all AI-polished text was checked against the original draft to confirm that no meaning or claims were altered during rewriting. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work did not involve a new human-subject study. Personalized memory can expose sensitive information or apply outdated preferences against a user's intent. Our forgetting analysis examines the second of these risks. A generated retraction record does not delete the underlying data or guarantee that an agent will stop using it. The VitaBench experiments run in benchmark simulations, and their results do not establish reliability for transactions in deployed services.

REPRODUCIBILITY STATEMENT

Section 3 describes the memory interface, inference procedure, and supervision. Section 4.1 summarizes the inference settings and training configuration, and Appendix A provides further implementation details and target construction.

REFERENCES

André Barreto, Vincent Dumoulin, Yiran Mao, Mark Rowland, Nicolas Perez-Nieves, Bobak Shahriari, Yann Dauphin, Doina Precup, and Hugo Larochelle. Capturing individual human preferences with reward features. *Advances in Neural Information Processing Systems*, 38:82119–82154, 2026.

James O Berger. Statistical decision theory. In *The New Palgrave Dictionary of Economics*, pp. 1–6. Springer, 1987.

Avinandan Bose, Zhihan Xiong, Yuejie Chi, Simon Shaolei Du, Lin Xiao, and Maryam Fazel. Lore: Personalizing llms via low-rank reward modeling. *arXiv preprint arXiv:2504.14439*, 2025.

Pierre Bourdieu. Outline of a theory of practice. In *The new social theory reader*, pp. 80–86. Routledge, 2020.

Pierre Bourdieu and Loïc JD Wacquant. *An invitation to reflexive sociology*. University of Chicago press, 1992.

Yuxin Chen, Yi Zhang, Zhengzhou Cai, Yaorui Shi, Zhiyuan Yao, Chenhang Cui, Jingnan Zheng, Yaqi Huo, Xi Su, Qi Gu, et al. Vitabench 2.0: Evaluating personalized and proactive agents in long-term user interactions. *arXiv preprint arXiv:2605.27141*, 2026.

Zihao Cheng, Hongru Wang, Zeming Liu, Yuhang Guo, Yuanfang Guo, Yunhong Wang, and Haifeng Wang. Toolspectrum: Towards personalized tool utilization for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20679–20699, 2025.

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.

Yupu Hao, Pengfei Cao, Zhuoran Jin, Huanxuan Liao, Yubo Chen, Kang Liu, and Jun Zhao. Evaluating personalized tool-augmented llms from the perspectives of personalization and proactivity. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21897–21935, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Wei-Chieh Huang, Weizhi Zhang, Yueqing Liang, Yuanchen Bei, Yankai Chen, Tao Feng, Zhen Tan, Yu Wang, Tianxin Wei, Shanglin Wu, et al. A survey of agent memory in the second half: Towards self-evolving and long-horizon agents. *Transactions on Machine Learning Research*, 2026.

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.

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.

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

Hannah Kim, Kushan Mitra, Chen Shen, Dan Zhang, and Estevam Hruschka. Aipom: Agent-aware interactive planning for multi-agent systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 85–96, 2025.

Mehmet Kurmaz. Aware-us: Preference-aware infeasibility resolution in tool-calling agents. *arXiv preprint arXiv:2601.02643*, 2026.

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

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. Hello again! llm-powered personalized agent for long-term dialogue. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5259–5276, 2025a.

Jia-Nan Li, Jian Guan, Wei Wu, and Rui Yan. Extended inductive reasoning for personalized preference inference from behavioral signals. *arXiv preprint arXiv:2505.18071*, 2025b.

Yi Li, Jieming Zhu, Weiwen Liu, Liangcai Su, Guohao Cai, Qi Zhang, Ruiming Tang, Xi Xiao, and Xiuqiang He. Pear: Personalized re-ranking with contextualized transformer for recommendation. In *Companion Proceedings of the Web Conference 2022*, pp. 62–66, 2022.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026.

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.

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

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.

Idan Shenfeld, Felix Faltings, Pulkit Agrawal, and Aldo Pacchiano. Language model personalization via reward factorization. *arXiv preprint arXiv:2503.06358*, 2025.

Yunxiao Shi, Wujiang Xu, Zhang Zeqi, Xing Zi, Qiang Wu, and Min Xu. Personax: A recommendation agent-oriented user modeling framework for long behavior sequence. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5764–5787, 2025.

Anikait Singh, Sheryl Hsu, Kyle Hsu, Eric Mitchell, Stefano Ermon, Tatsunori Hashimoto, Archit Sharma, and Chelsea Finn. Fspo: Few-shot optimization of synthetic preferences effectively personalizes to real users. In *International Conference on Learning Representations*, volume 2026, pp. 109496–109524, 2026.

Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40811–40838, 2026.

Yanming Wan, Jiaxing Wu, Marwa Abdulhai, Lior Shani, and Natasha Jaques. Enhancing personalized multi-turn dialogue with curiosity reward. *Advances in Neural Information Processing Systems*, 38:155857–155894, 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*, 2023.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems*, 37:53764–53797, 2024a.

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. *arXiv preprint arXiv:2402.04624*, 2024b.

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.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024c.

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

Manjie Xu, Xinyi Yang, Wei Liang, Chi Zhang, and Yixin Zhu. Inferring the Unspoken: Aligning Embodied Agents with Implicit Preferences, 2025a. URL https://arxiv.org/abs/2502.00858.

Qiancheng Xu, Yongqi Li, Heming Xia, Fan Liu, Min Yang, and Wenjie Li. Petoolllm: Towards personalized tool learning in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21488–21503, 2025b.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026a.

Yue Xu, Qian Chen, Zizhan Ma, Dongrui Liu, Wenxuan Wang, Xiting Wang, Li Xiong, and Wenjie Wang. Toward personalized llm-powered agents: Foundations, evaluation, and future directions. *arXiv preprint arXiv:2602.22680*, 2026b.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, 2026.

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. In *International Conference on Learning Representations*, volume 2026, pp. 39458–39486, 2026.

Weizhi Zhang, Xinyang Zhang, Chenwei Zhang, Liangwei Yang, Jingbo Shang, Zhepei Wei, Henry Peng Zou, Zijie Huang, Zhengyang Wang, Yifan Gao, et al. Personaagent: Bridging memory and action for personalized llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26421–26439, 2026.

Xinliang Frederick Zhang, Nick Beauchamp, and Lu Wang. Prime: Large language model personalization with cognitive dual-memory and personalized thought process. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33695–33724, 2025.

Xuan Zhang, Yang Deng, Zifeng Ren, See Kiong Ng, and Tat-Seng Chua. Ask-before-plan: Proactive language agents for real-world planning. In *Findings of the association for computational linguistics: EMNLP 2024*, pp. 10836–10863, 2024.

Jihao Zhao, Ding Chen, Zhaoxin Fan, Kerun Xu, Mengting Hu, Bo Tang, Feiyu Xiong, and Zhiyu Li. Inside out: Evolving user-centric core memory trees for long-term personalized dialogue systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13429–13446, 2026a.

Weixiang Zhao, Xingyu Sui, Yulin Hu, Jiahe Guo, Haixiao Liu, Biye Li, Yanyan Zhao, Bing Qin, and Ting Liu. Teaching language models to evolve with users: Dynamic profile modeling for personalized alignment. *Advances in Neural Information Processing Systems*, 38:33162–33199, 2026b.

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

A Implementation details

A.1 Inference and history selection

HabitusMem uses the same adapter on both benchmarks, with inference settings adapted to each benchmark (Section 4.1). Both evaluations use Hugging Face generation with greedy decoding and a serialized generation lock. The writer regenerates the memory whenever the history–request pair changes, so the memory can change during a VitaBench interaction without recursive updates to a persistent state. State-update and projection examples serve only as auxiliary supervision.

The VitaBench selector first reserves space for the instruction, the request suffix, and 32 additional tokens, and then combines an initial history segment, an elision marker, and the most recent interactions. The PersonaMem selector keeps the first 32,000 characters before the final token cutoff.

A.2 Parameter allocation

The base checkpoint is a vision-language model, and all inputs in our experiments are text. LoRA targets `q_proj`, `k_proj`, `v_proj`, `o_proj`, the linear-attention `in_proj_*` modules, and the feed-forward `gate_proj`, `up_proj`, and `down_proj`. Because the targeting rule matches all linear layers, 32.12M of the 205.23M adapter parameters (15.65%) go to visual projection and feed-forward modules; only the text path is used in our experiments. Embeddings and the output head are not trained.

A.3 Target construction and adapter selection

To construct auxiliary targets, we use 5,000-token history chunks, an 11,400-token input budget, teacher generation caps of 4,800 tokens for states and 1,800 for views, and final state and view budgets of 2,400 and 800 tokens. Action-projection examples are repeated three times in the final mixture. Target generation is retried when no syntactically complete record is produced. All examples come from the official training split, and direct writing uses its preference annotations as target supervision.

Action-projection examples take teacher-constructed states as inputs, whereas VitaBench inference takes raw history. The four tasks therefore teach one writer through complementary input formats, and their prompts do not all match the deployment prompt. A later data-builder option supports history-input action projections, but it was not used to train the reported adapter.

We selected the action weight on the 12-user development gate (166 subtasks, DeepSeek agent, shared execution guidance). The selected mixture reaches 0.1566 all-success, compared with 0.1205 for the uniform mixture and 0.1386 when direct writing is also weighted twice, which gives the highest average reward (0.3795 versus 0.3765); Rewrite reaches 0.0964 on the same gate. The uniform variant is estimated from available trials, and these comparisons do not cover the full 56-user benchmark. On the 5,000 PersonaMem-v2 benchmark queries, the selected and uniform adapters both reach 0.4788 MCQ accuracy. Only the selected mixture was evaluated on the full VitaBench benchmark.

B Baseline details

The ground-truth profile is a privileged reference, and full context gives the agent direct access to the history; neither is a deployable compact memory. RAG and Rewrite are implemented within the evaluation harness and represent specific versions of these approaches. On VitaBench 2.0, every memory system uses the same execution guidance and four trials per subtask on all 56 users. We do not evaluate query-aware compressors such as LongLLMLingua and RECOMP (Jiang et al., 2024; Xu et al., 2024) or a learned compressor trained on the same supervision.

C Diagnostics and qualitative analysis

C.1 Input budget and inference cost

Raising the writer input cap for 32k histories from 8,500 to 11,400 tokens changes MCQ accuracy by -0.0059 and does not change the head selector's truncation behavior; the main results use the original cap.

Writing memory for each distinct request requires another pass over the selected history. The 800-token cap bounds the generated records, while raw-history storage, writer prefill, latency, and service calls add further costs. We have not measured quality at matched inference cost across memory systems.

C.2 From memory constraints to completed actions

An analysis of an earlier writer examined 38 VitaBench subtasks on which it failed in both of two trials while full context succeeded in both. Among the resulting 76 failed trajectories, 55% stop before creating an order, 17% create an order without paying, and 28% pay with incorrect parameters. The corresponding memories contain the decisive constraints in 19 subtasks, contain them partially in 15, and miss them in four. Because the memory often held the needed constraint while the transaction still failed, we added action-oriented supervision that keeps exact choices and fulfillment constraints. This sample describes the earlier writer and does not estimate the failure rate of the final one.

C.3 Qualitative example of request relevance

PersonaMem-v2 persona 429 illustrates how a detailed background summary can omit a preference relevant to the current request. The user asks, "What are some relaxing outdoor activities I could do at home that might help me unwind after work?" A 1,935-character static memory lists occupational and educational background and several hobbies but omits the annotated preference for tending a small vegetable garden, and the same memory is reused for another request. The request-conditioned writer begins with "Enjoys gardening as a relaxing hobby", which the answer uses in its opening. This example comes from an earlier writer in the HabitusMem lineage and illustrates the motivation rather than the final model's performance.

D Bayesian interpretation of belief memory

Section 3.1 models the user as an evolving latent state and defines memory by posterior expected loss (Berger, 1987): the decision variable is a memory message, and its value depends on the actions it induces in a fixed agent. This appendix separates the evidential and decisional roles of the request, relates the formulation to habitus, and shows when conditioning the memory on the request can help.

D.1 The request as evidence and as decision

Let $\bar{b}_t(z_t) = \int p(z_t \mid z_{t-1})\, b_{t-1}(z_{t-1})\, dz_{t-1}$ denote the belief predicted from the interactions before the request. Setting $o_t = q_t$ in Equation 1 gives

$$b_t(z_t) \propto p(q_t \mid z_t)\, \bar{b}_t(z_t). \tag{5}$$

If the request carries no evidence about the user, its likelihood is constant in z_t and the belief equals the prediction from earlier interactions. The request still enters the loss $\ell_\tau(a, z_t; q_t)$ and can therefore change the ideal memory: a gardening preference can become central to an activity recommendation without becoming more credible because the user asks for an activity. A request that reveals a new circumstance, such as a changed delivery address, changes both the belief and the loss. In the implementation, input selection additionally determines which part of the history the writer sees.

D.2 Habitus and the latent state

Bourdieu describes habitus as durable dispositions, produced by past experience, that generate practices in concrete situations (Bourdieu, 2020), and as an open system of dispositions that new

experiences either reinforce or modify (Bourdieu & Wacquant, 1992, p. 133). In our formulation, the interaction history is the record of practices from which the assistant infers the user's state, the transition $p(z_t \mid z_{t-1})$ allows that state to change, and the scope field records the situations in which a disposition guides behavior. The computational state z_t is broader than habitus, since it also includes facts and explicit constraints that are not dispositions. The belief represents the assistant's uncertainty about the user rather than a model of the user's own reasoning, and our benchmarks do not test the sociological theory.

D.3 The value of conditioning memory on the request

When memories must fit a message budget, as in our 800-token implementation, conditioning on the request can only lower the best attainable risk. Fix a common history selection $\bar{H} = T(H)$, a task type τ, the agent and its guidance, and a budget B, and let $\mathcal{M}_B \subseteq \mathcal{M}$ contain the memories of at most B tokens. Write $\mathcal{R}_\tau(m; \bar{H}, Q)$ for the risk in Equation 2 given the selected history $\bar{H}$ and the request Q. Let $\mathcal{C}_B$ contain all encoders $c : \bar{H} \mapsto \mathcal{M}_B$, and let $\mathcal{W}_B$ contain all encoders $w : (\bar{H}, Q) \mapsto \mathcal{M}_B$. Assuming the expected risks exist, the best attainable decision risk satisfies

$$\inf_{w \in \mathcal{W}_B} \mathbb{E}_{\bar{H},Q} \left[\mathcal{R}_\tau(w(\bar{H}, Q); \bar{H}, Q) \right] \leq \inf_{c \in \mathcal{C}_B} \mathbb{E}_{\bar{H},Q} \left[\mathcal{R}_\tau(c(\bar{H}); \bar{H}, Q) \right]. \tag{6}$$

Proof. For every $c \in \mathcal{C}_B$, the encoder $w_c(h, q) = c(h)$ belongs to $\mathcal{W}_B$ and produces the same memory and risk. Taking the infimum over the larger class gives the inequality. The statement compares attainable risk with matched information, agent, and output budget; it does not imply that a trained writer reaches the infimum, and it ignores the cost of generating the memory.

A one-bit example shows that the inequality can be strict. Suppose the history contains independent uniform bits (U, V) and a uniform request asks for one of them. A request-conditioned encoder sends the requested bit and achieves zero error with a one-bit message. A static one-bit encoder, even with an optimal decoder, has minimum average error $1/4$: the four histories require four answer pairs, but one bit permits only two reconstruction pairs. At least two of the eight equally likely history–request cases are therefore answered incorrectly, and sending U attains the bound. The example isolates the benefit of allocating a limited message to a known request and does not model the benchmark distribution.

D.4 Textual records and the supervised approximation

A record $(\text{claim}, \text{evidence}, \text{scope}, \text{status})$ communicates a user-related assertion together with its source, applicability, and current standing. Its evidence field refers to an observation already present in the history and must not be counted as an independent observation in Equation 1. Likewise, the token probabilities in Equation 4 model textual continuations, not calibrated probabilities that the claims are true.

The status field distinguishes changes that a truth probability alone cannot express. A correction changes what the assistant believes happened, a preference update changes what is currently desired, and a request to stop using information restricts its use without necessarily making it false. Explicit records make these distinctions available to the agent, although correct use still depends on both the generated memory and the agent. Action-oriented supervision likewise consists of demonstrated textual memories whose effects are evaluated on downstream agent tasks; it does not train on observed user actions.