From Atom to Community: Structured and Evolving Agent Memory for User Behavior Modeling

Anonymous submission

Abstract

User behavior modeling lies at the heart of personalized applications such as recommender systems and intelligent assistants. With the rise of Large Language Model (LLM)-based agents, user preference representation has evolved from latent embeddings to semantic memory. However, understanding user preferences from implicit interactions remains challenging, as agents need to infer preferences from sparse behavioral signals such as clicks and purchases without explicit supervision. Current approaches typically rely on a single, unstructured summary to represent user preferences, updated through simple overwriting. However, this design is inherently suboptimal: users exhibit multi-faceted interests that get conflated in a single summary, preferences evolve over time yet naive overwriting causes interest forgetting, and individual interactions are often sparse, necessitating collaborative signals from similar users. To address this, we present *STEAM* (***ST****ructured and* ***E****volving* ***A****gent* ***M****emory*), a novel memory framework that reimagines how agent memory is organized and updated for user behavior modeling. *STEAM* decomposes user preferences into fine-grained atomic memory units, each capturing a distinct interest dimension with explicit links to observed behaviors. To exploit collaborative patterns, *STEAM* organizes semantically similar memories across users into communities and generates prototype memories for signal propagation. The framework further incorporates adaptive evolution mechanisms, including consolidation for refining existing memories and formation for capturing emerging interests. Experiments on three real-world recommendation datasets demonstrate that *STEAM* consistently improves recommendation accuracy, preference retention, and diversity.

1 Introduction

LLM-based agents (Masterman et al. 2024; Guo et al. 2024a) have become a promising paradigm for modeling users in personalized applications such as intelligent assistants (Li et al. 2024) and recommender systems (Huang et al. 2025a; Peng et al. 2025; Zhang et al. 2025a). A central capability of these agents is memory (Hu et al. 2025; Zhang et al. 2025b; Wu et al. 2025), which transforms historical interactions into persistent user representations that can be retrieved for future prediction and decision making. While existing memory studies primarily focus on explicitly stated information in dialogue (Lei, Li, and Zhang 2025; Tan et al. 2025; Zhong et al. 2024; Maharana et al. 2024; Yu et al. 2025), user behavior modeling requires agents to infer latent preferences from implicit signals such as clicks, ratings, and purchases.

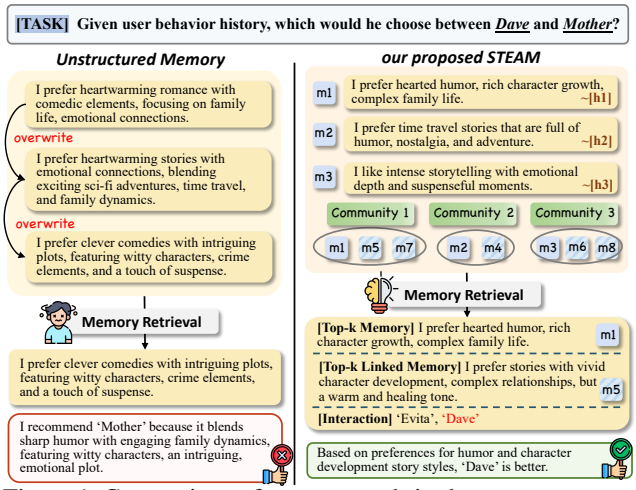

Figure 1: Comparison of unstructured single summary memory and our proposed *STEAM* for user behavior modeling in real recommendation scenario.

Unlike dialogue, user behaviors reveal preferences only implicitly. Existing user agents therefore infer latent interests from behavioral signals, use the resulting memory to predict future actions, and revise it through self-reflection on subsequent feedback. However, without explicit preference labels, accurate memory evolution remains difficult. As shown on the left of Fig. 1, representative methods (Zhang et al. 2024a,b; Cai et al. 2025; Guo et al. 2024b) typically compress user preferences into a single unstructured summary and rewrite it as new behaviors arrive. Repeated rewriting mixes newly observed interests with previous ones, leaving persistent, secondary, and emerging preferences entangled in the same representation. In Fig. 1, interests in family stories, time-travel adventures, and suspenseful comedies are gradually blended, allowing irrelevant evidence to interfere with retrieval and prediction. These limitations share a common cause: existing approaches treat the entire user profile as the basic unit of representation, evolution, and association, although different behaviors reflect different preference dimensions. This leads to three memory-specific challenges.

1) Disentangling Multi-Dimensional Preferences. Users

often exhibit multiple distinct or related interests. Compressing them into one summary blurs their boundaries and introduces irrelevant evidence during retrieval. Memory should instead represent coherent preference dimensions separately with their supporting behaviors. **2) Controllable Evolution of Dynamic Interests.** User interests may persist, strengthen, or emerge over time. Holistic rewriting can accumulate redundancy or obscure earlier preferences. Memory evolution should consolidate compatible evidence into existing units and create new units for emerging interests. **3) Exploiting Fine-Grained Collaborative Signals.** Sparse personal histories may provide insufficient evidence, while independently maintained summaries cannot connect related preference dimensions across users. As shown on the right of Fig. 1, linking atomic memories retrieves cross-user evidence about humor and character development, supporting the prediction. This motivates collaboration at the preference level rather than the whole-user level.

Accordingly, we replace the conventional single-summary paradigm with a compositional memory of fine-grained atomic units that collectively represent diverse user interests. We propose *STEAM* (***ST**ructured and **E**volving **A**gent **M**emory*), a self-evolving framework for user behavior modeling. As shown on the right of Fig. 1, *STEAM* links each memory unit to its supporting interactions and to cross-user memory communities of semantically related interests, enabling collaborative signal exploitation. Based on this structure, *STEAM* jointly supports community construction and memory evolution. It captures multi-hop memory connections to identify collaborative users and builds prototype memories for each community, enabling collaborative propagation without graph convolution. Meanwhile, it consolidates memories with similar interests and creates new units for emerging preferences. Experiments on three real-world datasets demonstrate the effectiveness of *STEAM* in recommendation, long-term preference retention, and diversity.

Our main contributions are summarized as follows:

- We propose atomic memories that disentangle heterogeneous user preferences and support independent, feedback-driven evolution.
- We organize cross-user atomic memories into prototype-guided communities, enabling structured multi-hop collaborative retrieval beyond direct semantic neighbors.
- Experiments on three datasets demonstrate consistent gains in recommendation effectiveness, long-term preference retention, and recommendation diversity.

2 Preliminary

2.1 Problem Formulation

Given a user set $\mathcal{U}$ and an item set $\mathcal{I}$, for each user $u \in \mathcal{U}$, a dedicated agent equipped with a memory module is assigned to track the user's evolving preferences. The **memory refinement phase** spans timestamps 1 to t, during which the agent iteratively predicts user behaviors and refines its memory through discrepancies between the prediction and the ground-truth. In the subsequent **inference phase**, the agent leverages the accumulated memory to generate personalized content (e.g., recommend items) tailored to the user's preferences. In this context, effective user preference modeling serves as the foundation for accurate next behavior prediction in user-centric applications.

2.2 Memory-Centric User Modeling Pipeline

Existing agent memory systems for user behavior modeling generally follow a two-stage pipeline consisting of memory refinement and inference. A brief review of representative memory architectures and LLM-based recommendation agents is provided in Appendix A.

• **Memory Refinement Phase.** Following (Zhang et al. 2024b; Cai et al. 2025), taking user u as an example, at each timestamp t, the assigned agent is tasked with simulating user u's selection between two candidate items i_{pos} and i_{neg}. The agent utilizes its memory m_t and current interaction records s_t to generate a predicted selection a_t:

$$a_t = \text{LLM}(m_t, s_t, i_{pos}, i_{neg}). \tag{1}$$

Through self-reflection, by analyzing the discrepancy between the user's actual choice i_{t+1} and predicted selection a_t, the agent refines its understanding of the user's preferences r_t, with its memory evolving accordingly.

$$r_t = \text{LLM}(a_t, i_{t+1}). \tag{2}$$

The entire interaction episode can thus be represented as:

$$\tau = \{(s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_t, a_t, r_t)\}, \tag{3}$$

through which the agent continuously adapts to dynamic user interests and progressively refines its memory.

• **Inference Phase.** For evaluation, existing methods formulate the task as a ranking problem to assess the fidelity of agentic user behavior modeling. Given a candidate set, the agent leverages refined memory and historical records to produce a personalized ranking. We evaluate performance using recommendation accuracy and diversity.

3 Methodology

To address agentic user behavior modeling, we propose *STEAM*, a systematic framework that represents user memory as structured units to disentangle diverse interests. It continuously refines memory through three operations: **Memory Formation** creates units for emerging interests; **Community Construction**, inspired by graph convolutional recommendation, identifies high-order neighbors aligned with fine-grained preferences; and **Memory Evolution** consolidates similar memories or forms new ones for novel preferences.

For next-behavior inference, *STEAM* retrieves memory from local to global, first querying relevant personal memories and then accessing collaborative memories through memory communities. During refinement, each agent alternates between forward prediction and backward reflection to evolve its memory; during inference, it uses the refined memory to generate personalized content without reflection.

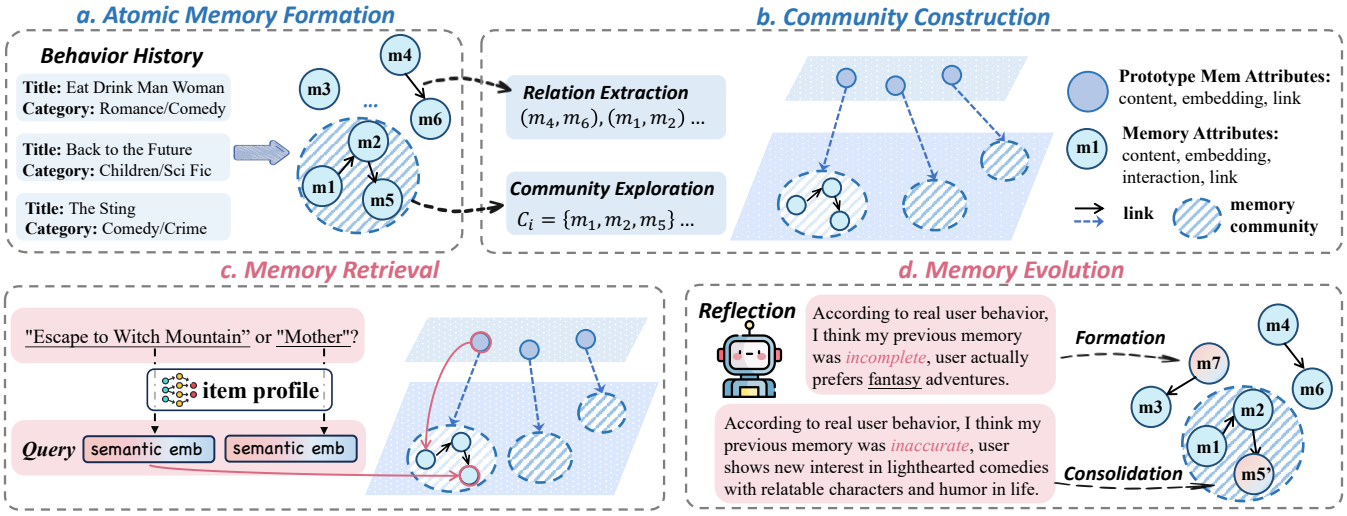

Figure 2: Overview of *STEAM*. (a–b) Memory construction: *Memory Formation* extracts preferences from user behavior, while *Community Construction* builds first-order relations, discovers communities via 2-hop BFS, and summarizes them as prototypes. (c) *Memory Retrieval* queries local memories before accessing collaborative ones through prototypes. (d) *Memory Evolution* reflects on simulated behaviors to consolidate existing memories or create new ones.

3.1 Memory Formation

We transform the memory paradigm in user behavior modeling into a set of atomic memory units. During self-reflection, if the agent identifies a new aspect of user interest, a new memory is formed. Each memory is a structured representation capturing a distinct aspect of user interest, consisting of content c, semantic embedding e, associated interaction behaviors I, and community link L:

$$\begin{aligned} m &= \{c, e, I, L\}, \\ e &= f_{\text{enc}}(c), \end{aligned} \tag{4}$$

where c describes the preference in natural language, e is its semantic embedding computed by a text encoder f_{enc}, I records interaction behaviors such as item clicks or purchases, and L stores the link to memory community.

For memory community discovery, we assign each memory a global index and store its semantic embedding in a shared space $\mathcal{M}$.

3.2 Community Construction

• **Motivation.** The expressiveness of individual memories is inherently constrained by sparse user interactions. To alleviate this limitation, we organize semantically related memory units across users into communities, allowing collaborative signals to be shared at the level of fine-grained preferences. Rather than treating each retrieved semantic neighbor independently, *STEAM* constructs a memory relation graph and explores bounded multi-hop neighborhoods.

• **Relation Extraction.** When a new memory m_i is formed, *STEAM* leverages its semantic embedding e_i to retrieve semantically related memories from the shared embedding space, which are regarded as *first-order collaborative neighbors*. For each memory $m_j \in \mathcal{M}$, we compute their cosine similarity, and then identify the top-k_1 most similar memories $\mathcal{N}_i$:

$$\mathcal{N}_i = \{\text{rank}(\text{sim}_{i,j}) \leq k_1,\ m_j \in \mathcal{M}\}, \tag{5}$$

where $\mathcal{N}_i$ excludes memory units belonging to the same user as m_i. We instantiate an undirected semantic relation between m_i and each memory in $\mathcal{N}_i$. Collectively, these relations form a cross-user memory graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes are atomic memories and the edges represent first-order semantic associations.

• **Community Exploration.** Relation extraction is performed online whenever a new memory is created, while community construction is triggered only after the current memory refinement batch is completed. Given a batch of newly formed memories $\{m_i\}_{i=1}^{B}$, their first-order semantic relations are first inserted into the existing memory graph. Direct semantic neighbors capture only pairwise similarity and may miss useful collaborative evidence connected through intermediate preference units. We therefore perform bounded graph exploration to identify structured multi-hop associations without treating all globally similar memories as independent evidence.

Specifically, we perform bounded 2-hop graph exploration using the newly inserted memories as exploration roots:

$$\mathcal{C}_i = \{m_j \in \mathcal{V} \mid d_{\mathcal{G}}(m_i, m_j) \leq 2\}, \tag{6}$$

where $d_{\mathcal{G}}(m_i, m_j)$ denotes the shortest-path distance between m_i and m_j in $\mathcal{G}$. We define $\mathcal{C}_i$ as the *2-hop memory community* rooted at m_i. Different communities may overlap, and a memory can participate in multiple communities when it connects several related preference dimensions. Such overlap preserves multiple collaborative contexts rather than enforcing a disjoint partition of the memory graph. Through this structure, collaborative retrieval provides access not only to semantically related preference descriptions but also to

their associated interaction attributes, offering item-level evidence for recommendation.

Each explored memory community is represented by a prototype memory p_k with summarized content c_k, semantic embedding e_k, and member links L_k:

$$p_k = \{c_k, e_k, L_k\}, \tag{7}$$

where $e_k = f_{\text{enc}}(c_k)$ and $L_k = \mathcal{C}_k$. Correspondingly, the link attribute of each member memory is updated to reference its prototype:

$$L_i \leftarrow L_i \cup \{p_k\}, \quad \forall m_i \in \mathcal{C}_k. \tag{8}$$

The prototype serves as a routing abstraction. During inference, it guides the agent toward relevant communities, from which candidate-relevant atomic memories and their supporting interactions are further selected as collaborative evidence. This structure connects users through shared preference dimensions while preserving fine-grained evidence for personalized prediction.

3.3 Memory Retrieval

STEAM operates in a local-to-global manner during memory retrieval. At local level, given a candidate item set I_c, the agent computes the semantic embeddings of each candidate and retrieves the top-k_2 relevant memories $\mathcal{M}_{\text{ret}}$ from its own memory module:

$$\mathcal{M}_{\text{ret}} = \{m_j \mid \text{rank}(\text{sim}_{c,j}) \leq k_2\}, \tag{9}$$

where $\text{sim}_{c,j}$ is the similarity between candidate item $i_c \in I_c$ and memory m_j. At the global level, the agent traces the link attribute of $\mathcal{M}_{\text{ret}}$ to locate prototype memories and explores their communities, leveraging similar preferences as collaborative signals. The retrieved local and global memories are then jointly exploited to generate a predicted selection.

3.4 Memory Evolution

After the agent generates recommendations based on retrieved memories, it reflects on the decision by comparing it with the real user behavior. This reflection triggers one of two evolution operations: *consolidation* or *formation*.

When a newly generated insight semantically overlaps with an existing memory, the agent performs *consolidation*: it leverages the LLM to merge their contents into a unified form, recomputes its semantic embedding, and appends the new interaction to its interaction attribute. The updated memory retains its link to the prototype memory. Through consolidation, redundant memories are eliminated, refining the overall preference representations.

If the reflection reveals a new aspect of user interest not captured by existing memories, the agent performs *formation* to construct a new memory, as detailed in Section 3.1. The complete procedure is summarized in Appendix B.

4 Experiments

4.1 Experimental Settings

• **Datasets.** We evaluate *STEAM* on three widely used recommendation datasets covering diverse domains: Amazon CDs, Amazon Clothing, and MovieLens-100K. For all datasets, we use the data processing and evaluation splits by RecBole (Zhao et al. 2022) to ensure fair comparison. Table 1 summarizes their statistics, with detailed dataset descriptions and preprocessing procedures provided in Appendix C.1.

Datasets	#Users	#Items	#Inters.	Sparsity
CDs	112,135	73,304	1,441,330	99.982%
Clothing	443,895	353,709	5,956,426	99.996%
ML-100K	943	1,341	98,996	92.172%

Table 1: Statistics of the preprocessed datasets

• **Baselines.** We compare *STEAM* with the following recommendation methods, grouped by user modeling approach: 1) Without User Representation. **Pop** recommends by item popularity; **BM25** (Robertson and Zaragoza 2009) ranks candidates by textual similarity to interacted items. 2) Latent Embedding Methods. **BPR** (Rendle et al. 2012) optimizes user/item embeddings via BPR loss; **SASRec** (Kang and McAuley 2018) captures sequential patterns through self-attention. 3) LLM-based Semantic Modeling. **LLMRank** (Hou et al. 2024) verbalizes user behavior sequences and ranks via LLMs' contextual understanding; **InteRecAgent** (Huang et al. 2025b) extracts and updates user profiles from conversation history; **AgentCF** (Zhang et al. 2024b) employs user and item agents to preserve preferences and collaborative signals, updating memory via simulated interactions; **AFL** (Cai et al. 2025) refines the agent's understanding of user preferences through a feedback loop between user and recommender agents.

• **Implementation Details.** We use *GPT-3.5-Turbo* as the default backbone for all LLM-based methods, while results with additional backbones (*Claude-Haiku-4.5*, *Llama-3-70B*, and *GPT-5.1*) are reported in Appendix D, demonstrating consistent performance trends across different LLMs. The hyper parameter k_1=2, k_2=1, and *all-MiniLM-L6-v2* is used for semantic retrieval. We discuss the hyperparameter sensitivity in Section 4.6. Following (Cai et al. 2025), we utilize a randomly sampled subset of 1000 users for Amazon and all users for ML-100K in subsequent studies. More comprehensive implementation details are provided in Appendix C.2.

4.2 Recommendation Evaluation

• **Evaluation Settings.** Since the quality of learned user memory is not directly observable, we evaluate it through downstream recommendation. Following (Zhang et al. 2024b; Cai et al. 2025), each LLM-based agent ranks a shared candidate set containing the ground-truth next item and nine randomly sampled non-interacted items. Pop, BM25, BPR_{full}, and $\text{SASRec}_{\text{full}}$ instead rank the full item catalog and are reported only as protocol-different references. For InteRecAgent, each ranking instance is formulated as a single recommendation turn following its original protocol. We report *NDCG@K* for $K \in \{1, 5, 10\}$.

• **Recommendation Performance.** The results are reported in Table 2. 1) **Impact of Sparse Supervision.** The substantial degradation from full-data BPR and SASRec to their

Method	CDs			Clothing			ML-100K		
	N@1	N@5	N@10	N@1	N@5	N@10	N@1	N@5	N@10
Pop†	0.1037	0.2359	0.4217	0.0787	0.1321	0.3794	0.0743	0.1192	0.3718
BM25†	0.1267	0.2197	0.4176	0.0947	0.1511	0.3892	0.0647	0.1062	0.3658
BPR$^{\dagger}_{\text{full}}$	0.1754	0.4191	0.5236	0.2154	0.3312	0.4748	0.1053	0.2143	0.4172
SASRec$^{\dagger}_{\text{full}}$	0.3509	0.6857	0.7248	0.3804	0.4896	0.5906	0.1148	0.6404	0.6573
BPR$_{\text{sample}}$	0.1127	0.3101	0.4328	0.1043	0.2131	0.4094	0.1007	0.1814	0.4019
SASRec$_{\text{sample}}$	0.1938	<u>0.4051</u>	0.4937	<u>0.2164</u>	0.3152	0.4826	<u>0.1627</u>	0.3136	0.4618
LLMRank	0.0507	0.1789	0.3832	0.0604	0.1902	0.3917	0.0509	0.2013	0.3981
InteRecAgent	0.0608	0.2199	0.4054	0.1606	0.3414	0.4639	0.1044	0.1601	0.3956
AgentCF	0.1706	0.3927	0.5098	0.1704	0.3572	0.5071	0.1406	0.3261	<u>0.4804</u>
AFL	<u>0.2107</u>	0.3916	<u>0.5281</u>	0.2108	<u>0.3909</u>	<u>0.5132</u>	0.1509	<u>0.3314</u>	0.4792
STEAM (Ours)	**0.3270***	**0.5321***	**0.6004***	**0.2860***	**0.5332***	**0.6151***	**0.1780***	**0.4068***	**0.5087***
Rel. Improv.	*+55.20%*	*+31.35%*	*+13.69%*	*+32.16%*	*+36.40%*	*+19.86%*	*+9.40%*	*+22.75%*	*+5.89%*

Table 2: Recommendation performance comparison. N@K denotes NDCG@K. Among methods evaluated on the shared 10-item candidate sets, the best and second-best results are highlighted in **bold** and <u>underlined</u>, respectively. Methods marked with † are evaluated under the full-catalog protocol and are included only as references. Relative improvements are computed against the strongest sampled-data baseline for each metric. * denotes a statistically significant improvement over that baseline under paired testing ($p < 0.05$).

Stage	Metric	CDs			Clothing			ML-100K		
		GPT-3.5-Turbo	AFL	Ours	GPT-3.5-Turbo	AFL	Ours	GPT-3.5-Turbo	AFL	Ours
Early	Precision	0.1693	<u>0.3279</u>	**0.3462**	0.1598	<u>0.1620</u>	**0.3724**	0.0919	<u>0.0958</u>	**0.2044**
	F1	0.2481	<u>0.3423</u>	**0.3847**	<u>0.2695</u>	0.2678	**0.4945**	<u>0.1740</u>	0.1698	**0.2948**
Middle	Precision	0.2260	<u>0.3679</u>	**0.3798**	0.2057	<u>0.2095</u>	**0.3457**	0.1478	<u>0.1526</u>	**0.2229**
	F1	0.3169	<u>0.3944</u>	**0.4219**	0.3132	<u>0.3216</u>	**0.4673**	0.2356	<u>0.2372</u>	**0.3154**
Recent	Precision	0.2827	<u>0.4079</u>	**0.4107**	0.2516	<u>0.2570</u>	**0.2880**	0.2037	<u>0.2094</u>	**0.2254**
	F1	0.3857	<u>0.4465</u>	**0.4528**	0.3569	<u>0.3754</u>	**0.4086**	0.2972	<u>0.3046</u>	**0.3225**

Table 3: Preference retention performance across early, middle, and recent interaction stages. Bold and underlined values denote the best and second-best results, respectively.

sampled-data variants demonstrates the sensitivity of ID-based recommenders to limited interaction supervision. In contrast, LLM-based agents can exploit item semantics in addition to behavioral supervision, which may partly explain their stronger performance in the sampled setting. 2) **Effectiveness of Memory Refinement.** AgentCF, AFL, and *STEAM* generally outperform LLMRank, which directly ranks candidates from raw interaction histories. This suggests that explicitly refining and retrieving user memory can better consolidate preference evidence across interactions than relying solely on in-context histories. 3) ***STEAM* Performance.** Under the sampled-data setting, *STEAM* achieves the best performance on all nine metrics. The gains are generally larger on the two sparse Amazon datasets, indicating that structured atomic memories and cross-user collaborative signals are particularly useful when personal interaction supervision is limited. The contribution of each component is further examined in the ablation study.

4.3 Long-term Preference Retention Evaluation

• **Motivation.** While next-item recommendation evaluates subsequent behavior prediction, it does not directly measure whether the memory preserves preferences expressed at different stages of interaction history. Following the binary preference-discrimination protocol adopted in prior user-agent studies (Zhang et al. 2024a; Cai et al. 2025), we evaluate whether the final memory retains preferences expressed during the early, middle, and recent stages of user interaction.
• **Evaluation Settings.** Each user sequence is divided into five temporal intervals, with one positive interaction held out from each interval before memory refinement. Intervals 1–2, 3, and 4–5 define the *Early*, *Middle*, and *Recent* stages, respectively. For each held-out positive, we sample nine non-interacted items to form a 10-item candidate set. The agent independently predicts whether the user would prefer each candidate, from which Precision and F1 are computed. Candidate sets and evaluation instructions are fixed across methods. The metrics are first computed for each interval and then macro-averaged within each temporal stage.

• **Preference Retention Performance.** Table 3 shows that *STEAM* outperforms GPT-3.5-Turbo and AFL across all datasets, temporal stages, and metrics. The gains are particularly pronounced for early-stage preferences on Clothing and ML-100K, indicating that structured atomic memories better preserve preference evidence expressed earlier in the interaction history. Moreover, *STEAM* exhibits smaller or comparable F1 variation between the early and recent stages, suggesting more balanced retention across time rather than a bias toward only the latest interactions.

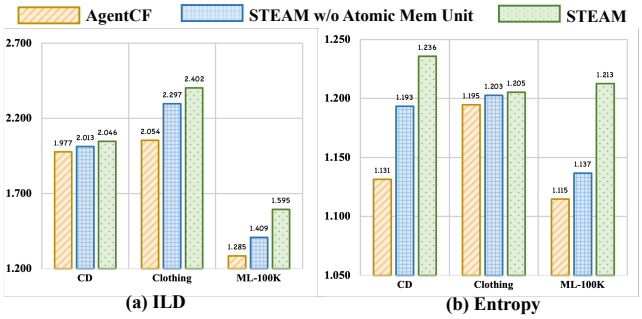

Figure 3: Recommendation diversity under the 100-item candidate setting. Panels (a) and (b) report ILD@5 and Entropy@5, respectively.

4.4 Diversity Evaluation

• **Evaluation Settings.** For each user, we construct a candidate pool consisting of the held-out positive item and 99 randomly sampled non-interacted items shared across all methods. Following (Yin et al. 2023), we evaluate the top-5 recommendations using two complementary metrics: *Intra-List Distance (ILD)* and *Entropy*. ILD measures the average semantic dissimilarity among recommended items, while Entropy measures the diversity and balance of their category distribution. We compare *STEAM* with AgentCF and *STEAM* w/o AMU, where atomic memory units are replaced with a single user summary while all other procedures remain unchanged. AFL is excluded because it relies on an external recommender to generate candidate lists, making its diversity dependent on a separately trained model.

• **Diversity Performance.** While Table 2 demonstrates the strong recommendation effectiveness of *STEAM* under the shared candidate-ranking protocol, Fig. 3 shows that *STEAM* also achieves consistently higher diversity under the more challenging 100-item candidate setting. Meanwhile, *STEAM* maintains competitive recommendation accuracy, indicating that the diversity gains are not achieved at the expense of recommendation relevance. It obtains the highest ILD@5 and Entropy@5 on all three datasets, indicating greater semantic variation and a more balanced category distribution. Replacing atomic memory units with a single user summary consistently reduces both diversity metrics and recommendation effectiveness, with particularly clear Entropy degradation on Clothing and ML-100K, while still outperforming AgentCF. These results suggest that fine-grained atomic memory representation contributes to both recommendation quality and diversity: preserving distinct preference dimensions enables the agent to retrieve evidence from multiple interests and produce relevant lists with broader semantic and categorical coverage.

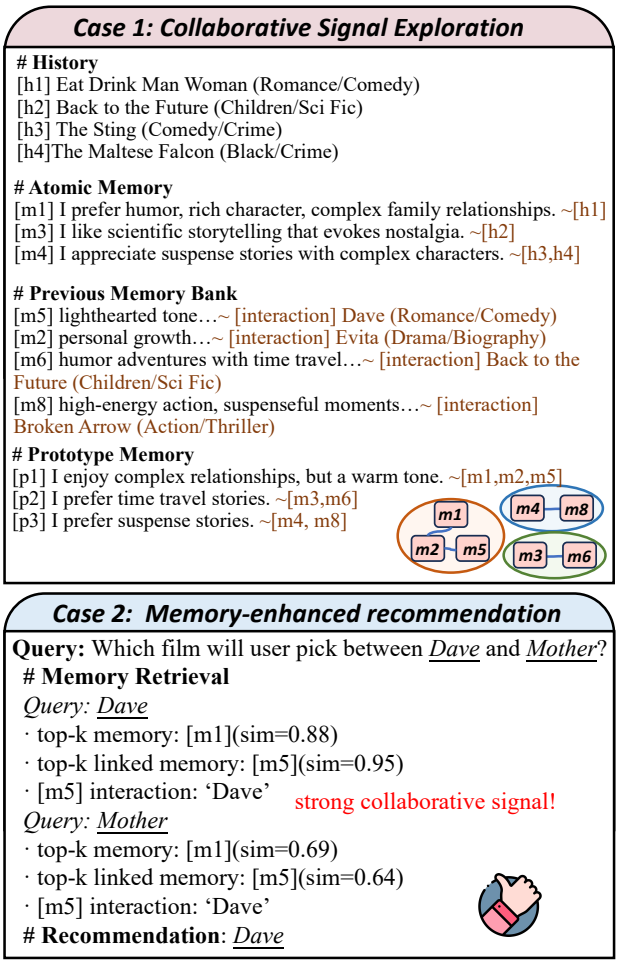

Figure 4: Case study of case1: collaborative memory construction and case2: collaborative evidence retrieval for recommendation.

4.5 Case Study

To intuitively examine how *STEAM* models and leverages collaborative signals, we present two cases in Fig. 4. In **Case 1**, we illustrate preference modeling for the interaction sequence $\{h_1, h_2, h_3, h_4\}$. During atomic memory formation, h_3 and h_4 both express a preference for suspense stories in the crime genre, and m_4 consolidates their compatible evidence. During community exploration, (m_1, m_2) and (m_2, m_5) form first-order associations. A 2-hop BFS therefore identifies the community $\{m_1, m_2, m_5\}$, which is summarized into a prototype while retaining links to its constituent atomic memories. In **Case 2**, the agent predicts whether the user will choose *Dave* or *Mother*. Personal retrieval first identifies m_1, but this evidence alone provides limited candidate-specific support. Through the associated

Method	CDs			Clothing			ML-100K		
	N@1	N@5	N@10	N@1	N@5	N@10	N@1	N@5	N@10
STEAM	0.3270	0.5321	0.6004	0.2860	0.5332	0.6151	0.1780	0.4068	0.5087
STEAM w/o AMU	0.2300	0.4947	0.5765	0.2700	0.5005	0.5782	0.1300	0.3404	0.4796
STEAM w/o ME	0.2400	0.4973	0.5835	0.2280	0.4958	0.5635	0.1350	0.3287	0.4785
STEAM w/o MC	0.2600	0.5017	0.5886	0.2300	0.4966	0.5640	0.1400	0.3295	0.4788
STEAM w/o CC	0.2800	0.4992	0.5906	0.2600	0.5023	0.5790	0.1600	0.3691	0.5060
STEAM w/ Direct-kNN	0.3000	0.5146	0.5948	0.2750	0.5183	0.5957	0.1450	0.3554	0.4913

Table 4: Ablation results on three datasets. N@K denotes NDCG@K.

prototype p_1, the agent further retrieves cross-user atomic memory m_5, which is unavailable from personal memory alone and has the highest semantic similarity to *Dave*. Its interaction attribute provides candidate-consistent behavioral evidence, helping resolve the ambiguity and supporting the final selection of *Dave*. This example illustrates how structured community retrieval supplements personal memory with relevant collaborative evidence rather than merely retrieving independent semantic neighbors.

4.6 In-depth Analysis

• **Impact of Different Components of *STEAM*.** We evaluate five variants covering memory representation, evolution, and collaborative retrieval. 1) *STEAM* w/o AMU: atomic memories in Section 3.1 are replaced with a single user summary. 2) *STEAM* w/o ME: feedback-driven evolution is removed, leaving the initialized memory bank fixed. 3) *STEAM* w/o MC: new interests are appended without consolidating overlapping units. 4) *STEAM* w/o CC: community construction and cross-user retrieval in Section 3.2 are removed, retaining only personal memories. 5) *STEAM* w/ Direct-kNN: community construction is replaced by direct retrieval of semantically similar cross-user atomic memories under the same evidence budget.

Table 4 shows that component contributions vary across datasets. Replacing atomic memories with a single summary substantially degrades performance, especially on CDs and ML-100K, supporting fine-grained preference representation. Removing memory evolution or consolidation also consistently hurts performance: the former prevents emerging preferences from being incorporated, while the latter retains overlapping units that may introduce redundant or conflicting evidence. This supports that evolution and consolidation jointly maintain coherent user representations. Community construction generally brings clearer gains on Amazon datasets. Direct-kNN improves over personal-only retrieval on these datasets but remains below the full *STEAM*, showing that communities and prototypes organize collaborative evidence more effectively than independent semantic neighbors. On ML-100K, community removal has a smaller effect, whereas Direct-kNN underperforms personal-only retrieval, suggesting that unstructured cross-user evidence may add noise when personal signals are sufficient. Overall, atomic representation, controlled evolution, and structured community retrieval provide complementary benefits.

• **Impact of Hyperparameters.** We analyze k_1, the number of global atomic memories considered for relation extraction in Eq. (5), and k_2, the number of local memories retrieved for each candidate. Figure 5 reports the results on CDs and Clothing due to space constraints, with consistent trends observed across both datasets and $k_1 = 2, k_2 = 1$ achieving the best performance among the tested settings. A small k_1 fails to capture sufficient cross-user relations, limiting the coverage of collaborative evidence, whereas a large value introduces noisy edges that weaken the semantic coherence of memory communities. Similarly, an overly large k_2 may retrieve weakly related personal preferences and introduce distracting evidence into the LLM context. These results indicate that collaborative retrieval benefits more from compact and precise evidence than from simply expanding the retrieved context.

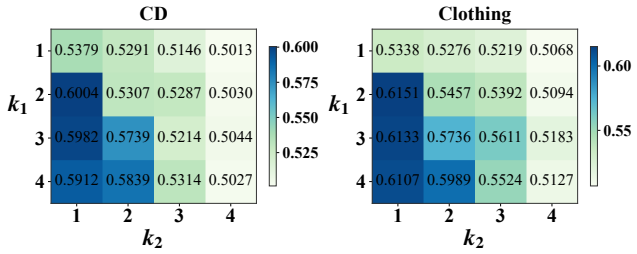

Figure 5: Sensitivity to k_1 and k_2 measured by NDCG@10.

5 Conclusion

In this paper, we focus on agent memory mechanisms for user behavior modeling. To address key challenges of preference diversity, dynamics, and sparsity, we transform single summary memory into fine-grained atomic memory units. We propose *STEAM*, a self-evolving memory framework that consolidates and evolves atomic memories to capture diverse and evolving preferences, while organizing them into communities with prototype memories to model collaborative signals. Experiments on recommendation tasks validate the effectiveness of *STEAM*. We hope this work inspires further research on structured memory mechanisms for effective LLM-based agents in user behavior modeling and beyond.

References

Cai, S.; Zhang, J.; Bao, K.; Gao, C.; Wang, Q.; Feng, F.; and He, X. 2025. Agentic feedback loop modeling improves recommendation and user simulation. In *Proceedings of the 48th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2235–2244.

Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N. V.; Wiest, O.; and Zhang, X. 2024a. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Guo, T.; Liu, C.; Wang, H.; Mannam, V.; Wang, F.; Chen, X.; Zhang, X.; and Reddy, C. K. 2024b. Knowledge graph enhanced language agents for recommendation. *arXiv preprint arXiv:2410.19627*.

Hou, Y.; Zhang, J.; Lin, Z.; Lu, H.; Xie, R.; McAuley, J.; and Zhao, W. X. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, 364–381. Springer.

Hu, Y.; Liu, S.; Yue, Y.; Zhang, G.; Liu, B.; Zhu, F.; Lin, J.; Guo, H.; Dou, S.; Xi, Z.; et al. 2025. Memory in the Age of AI Agents. *arXiv preprint arXiv:2512.13564*.

Huang, C.; Wu, J.; Xia, Y.; Yu, Z.; Wang, R.; Yu, T.; Zhang, R.; Rossi, R. A.; Kveton, B.; Zhou, D.; et al. 2025a. Towards agentic recommender systems in the era of multimodal large language models. *arXiv preprint arXiv:2503.16734*.

Huang, X.; Lian, J.; Lei, Y.; Yao, J.; Lian, D.; and Xie, X. 2025b. Recommender ai agent: Integrating large language models for interactive recommendations. *ACM Transactions on Information Systems*, 43(4): 1–33.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Lei, X.; Li, Q.; and Zhang, M. 2025. D-SMART: Enhancing LLM Dialogue Consistency via Dynamic Structured Memory And Reasoning Tree. *arXiv preprint arXiv:2510.13363*.

Li, Y.; Wen, H.; Wang, W.; Li, X.; Yuan, Y.; Liu, G.; Liu, J.; Xu, W.; Wang, X.; Sun, Y.; et al. 2024. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*.

Maharana, A.; Lee, D.-H.; Tulyakov, S.; Bansal, M.; Barbieri, F.; and Fang, Y. 2024. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*.

Masterman, T.; Besen, S.; Sawtell, M.; and Chao, A. 2024. The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. *arXiv preprint arXiv:2404.11584*.

Peng, Q.; Liu, H.; Huang, H.; Yang, Q.; and Shao, M. 2025. A survey on llm-powered agents for recommender systems. *arXiv preprint arXiv:2502.10050*.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Robertson, S.; and Zaragoza, H. 2009. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc.

Tan, Z.; Yan, J.; Hsu, I.-H.; Han, R.; Wang, Z.; Le, L.; Song, Y.; Chen, Y.; Palangi, H.; Lee, G.; et al. 2025. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8416–8439.

Wu, Y.; Liang, S.; Zhang, C.; Wang, Y.; Zhang, Y.; Guo, H.; Tang, R.; and Liu, Y. 2025. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *arXiv preprint arXiv:2504.15965*.

Yin, Q.; Fang, H.; Sun, Z.; and Ong, Y.-S. 2023. Understanding diversity in session-based recommendation. *ACM Transactions on Information Systems*, 42(1): 1–34.

Yu, S.; Cheng, M.; Wang, D.; Liu, Q.; Liu, Z.; Guo, Z.; and Tao, X. 2025. MemWeaver: A Hierarchical Memory from Textual Interactive Behaviors for Personalized Generation. *arXiv preprint arXiv:2510.07713*.

Zhang, A.; Chen, Y.; Sheng, L.; Wang, X.; and Chua, T.-S. 2024a. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, 1807–1817.

Zhang, J.; Hou, Y.; Xie, R.; Sun, W.; McAuley, J.; Zhao, W. X.; Lin, L.; and Wen, J.-R. 2024b. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM Web Conference 2024*, 3679–3689.

Zhang, Y.; Qiao, S.; Zhang, J.; Lin, T.-H.; Gao, C.; and Li, Y. 2025a. A survey of large language model empowered agents for recommendation and search: Towards next-generation information retrieval. *arXiv preprint arXiv:2503.05659*.

Zhang, Z.; Dai, Q.; Bo, X.; Ma, C.; Li, R.; Chen, X.; Zhu, J.; Dong, Z.; and Wen, J.-R. 2025b. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6): 1–47.

Zhao, W. X.; Hou, Y.; Pan, X.; Yang, C.; Zhang, Z.; Lin, Z.; Zhang, J.; Bian, S.; Tang, J.; Sun, W.; Chen, Y.; Xu, L.; Zhang, G.; Tian, Z.; Tian, C.; Mu, S.; Fan, X.; Chen, X.; and Wen, J. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. In *CIKM*, 4722–4726. ACM.

Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; and Wang, Y. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19724–19731.