AtomRec: Evolving Atomic Memory for Agentic Recommendation

Anonymous ACL submission

Abstract

Agentic recommender systems use large language models to maintain semantic memory and support evidence-aware recommendation. However, existing memory mechanisms often compress user and item information into coarse summaries and connect them with scalar collaborative links, making it difficult to preserve fine-grained preference stages or retrieve interpretable evidence as user interests evolve. We propose ATOMREC, an agentic recommender with evolving atomic collaborative memory. ATOMREC represents user and item memories as structured atomic units, builds semantic links across related memories, and evolves related historical fields when new interactions arrive. During recommendation, it retrieves linked memories as multi-hop evidence paths rather than isolated neighbor summaries, allowing collaborative signals to support grounded ranking. Experiments on four public benchmarks show that ATOMREC consistently outperforms state-of-the-art agentic and memory-augmented baselines, with around 8.5% average relative improvement across metrics. Code is included in the supplementary materials.

1 Introduction

Recommender systems are moving from representation-based prediction toward agentic recommendation (Ko et al., 2022; Da'u and Salim, 2020; Deldjoo et al., 2024; Peng et al., 2025; Zhang et al., 2025a). Traditional models encode user preferences with rating matrices, collaborative embeddings, or sequential hidden states (Kang and McAuley, 2018; Sun et al., 2019). In contrast, LLM-powered recommender agents need to understand user intent, maintain long-term memory, integrate collaborative evidence, and support evidence-aware ranking (Wu et al., 2024; Liu et al., 2024). Memory therefore becomes a key interface between historical interactions, collaborative signals, and language-based reasoning (Hu et al., 2025).

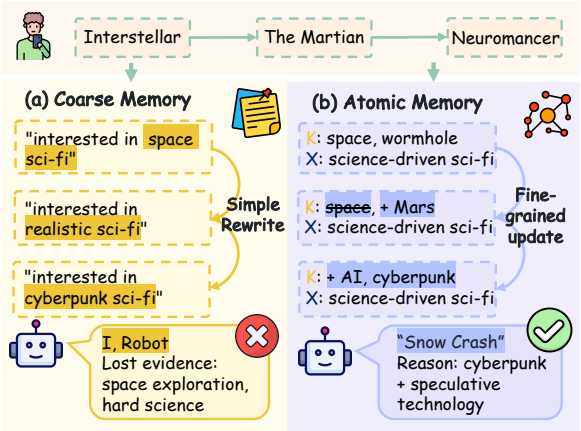

Figure 1: Motivation of ATOMREC. Given the same evolving user history, coarse memory rewrites broad summaries and may lose fine-grained preference evidence, while atomic memory preserves preference traces as operational units for field/link evolution and evidence-path retrieval.

In long-term recommendation, memory often needs to be fine-grained and dynamic (Ko et al., 2022; Da'u and Salim, 2020). A user's preference is not a single stable profile, but a collection of small and evolving traces, such as recurring genres, recent intents, item attributes, and collaborative signals from related users or items (Ko et al., 2022; Da'u and Salim, 2020). These traces may evolve at different speeds: some reflect stable long-term interests, while others capture emerging or context-dependent preferences. As illustrated in Figure 1, a user may move from space exploration in *Interstellar*, to realistic survival science fiction in *The Martian*, and later to cyberpunk society in *Neuromancer*. These stages remain related under the broad science-fiction theme, but each carries different recommendation evidence.

Recent agentic recommender systems use semantic memory to store user preferences and item prop-

erties, while collaborative memory, planner-based, and propagation-based agents further improve recommendation through user–item memory graphs, tool routing, information-gap analysis, or preference propagation (Chen et al., 2026; Xu et al., 2025; Zhang et al., 2024b; Li et al., 2026b; Zhang et al., 2026a; Li et al., 2026a). Despite this progress, existing memory mechanisms still face two key limitations. **First, coarse granularity.** Summary-level memories compress multiple preference traces into one profile, making selective revision difficult. **Second, limited semantic operability.** Scalar interaction, similarity, or propagation weights capture relational strength, but provide little evidence about why memories are related (Peng et al., 2025; Zhu et al., 2025).

These limitations are especially harmful for long-term recommendation. As shown in Figure 1, coarse updates may repeatedly rewrite a broad user profile, causing earlier fine-grained evidence, such as space exploration or hard-science survival, to be weakened by later interests. Meanwhile, retrieval often returns isolated neighbor summaries rather than evidence paths that explain how earlier preferences connect to emerging interests. As a result, the recommender agent may receive collaborative context, but still lack fine-grained and interpretable memory evidence for ranking.

To address the above limitations, inspired by recent atomic memory systems (Xu et al., 2026; Chhikara et al., 2025), we propose ATOMREC, an LLM-based agentic recommender system with evolving atomic collaborative memory. The key idea is to make recommendation memory reorganizable rather than merely more detailed. ATOMREC replaces coarse entity-level summaries with atomic memory units containing structured semantic fields, which serve as operational objects for locating, linking, revising, and preserving specific preference traces.

On top of these atomic units, ATOMREC introduces Semantic Collaborative Link Construction and dynamic memory evolution. The linking module builds semantic relations across user and item atoms, turning scalar collaborative signals into interpretable evidence. The evolution module updates related historical atoms and their links when new interactions provide additional evidence, enabling the memory graph to reorganize over time instead of passively accumulating summaries. During recommendation, ATOMREC retrieves along the evolved links to construct context-aware evidence paths for final ranking.

Our contributions are as follows:

1. We propose ATOMREC, an agentic recommender system that represents user and item memories as linked and evolving atomic units.

2. We introduce Semantic Collaborative Link Construction and dynamic memory evolution to support fine-grained semantic reorganization and context-aware evidence retrieval.

3. We conduct experiments on four public benchmarks, showing that ATOMREC outperforms strong baselines with around 8.5% average relative improvement and better handles users with stronger preference drift.

2 Method

2.1 Problem Formulation

Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets. For a target user u at time t, we denote the historical interactions as $\mathcal{H}_u^{t-1}$ and the candidate set as $\mathcal{C}_u^t \subseteq \mathcal{I}$. Given a natural language instruction I_u, the task is to rank candidates according to the user's current preference.

ATOMREC maintains a memory space $\mathcal{M}$ of user and item memories, where each memory is an evolvable atomic note and semantic links connect related notes. Let M_u^t denote the pre-t user memory, M_i^t the memory of candidate item i, and M_{collab}^t the collaborative evidence retrieved from $\mathcal{M}$. The recommendation agent scores each candidate by

$$r_{u,i}^t = \text{LM}_{\text{Rec}}\left(I_u, M_u^t, M_{\text{collab}}^t, M_i^t\right), \quad i \in \mathcal{C}_u^t.$$

Candidates are ranked by $r_{u,i}^t$. All memory construction, linking, evolution, and retrieval use only interactions before time t; the target item is used only as a candidate during final reranking and is never written back into memory for the same test instance.

2.2 Overview

Recommendation memory contains heterogeneous preference traces, such as stable interests, recent intents, item attributes, and collaborative signals. Existing memory-based and agentic recommenders often compress these traces into coarse user/item summaries and connect them with interaction, similarity, or propagation weights (Chen et al., 2026; Xu et al., 2025; Tsitseklis et al., 2023; Zhang et al.,

2024b; Peng et al., 2025; Zhu et al., 2025). This can obscure preference stages and provide limited semantic evidence for selective revision and evidence-aware retrieval.

To address these limitations, we propose ATOMREC, an agentic recommender with evolving atomic collaborative memory. As shown in Figure 2, ATOMREC follows four steps. First, **atomic collaborative memory construction** represents user and item memories as structured notes, providing fine-grained units for selective retrieval and revision. Second, **Semantic Collaborative Link Construction** converts nearby memories into semantic relations, making collaborative links explainable rather than only numerical. Third, **dynamic memory evolution** updates related historical notes when new evidence arrives, allowing memory to track emerging interests while preserving temporal traces. Finally, **context-aware collaborative retrieval** expands along semantic links and synthesizes multi-hop evidence paths for grounded ranking.

2.3 Atomic Collaborative Memory Construction

ATOMREC represents each user or item memory as an atomic note rather than a single coarse summary. For a user u at time t, the atomic memory note is defined as

$$m_u^t = \{c_u^t, t_u, K_u^t, G_u^t, X_u^t, e_u^t, L_u^t\},$$

where c_u^t denotes the textual memory content, t_u is the timestamp, K_u^t is a set of keywords, G_u^t is a set of semantic tags, X_u^t is a short contextual description, e_u^t is a dense embedding, and L_u^t stores linked memory identifiers. Item memories follow the same structure. The embedding of each memory note is computed from its semantic fields:

$$e_u^t = f_{\text{enc}}\left(\text{concat}(c_u^t, K_u^t, G_u^t, X_u^t)\right),$$

where f_{enc} is a text encoder.

This structured note design makes memory both retrievable and interpretable: the content field preserves original preference evidence, keywords and tags expose explicit semantic facets, the context field summarizes current intent, and the link field allows the note to participate in the collaborative memory graph. We do not assume that these fields are fully disentangled; instead, we use them as complementary semantic views that make retrieval, linking, and evolution more controllable. Compared with coarse user profiles, atomic notes reduce the tendency to collapse heterogeneous interests into a single summary and provide a finer-grained substrate for selectively revising or retrieving preference evidence.

2.4 Semantic Collaborative Link Construction

Traditional collaborative memory usually connects users and items through predefined interaction edges, co-occurrence statistics, or scalar similarity weights. These edges indicate collaborative strength, but they do not explain the semantic relation between memories. ATOMREC instead builds semantic links among atomic memory notes.

When a new memory note m_n is created, we first retrieve semantically nearby memories from the memory space $\mathcal{M}$ using embedding similarity:

$$s_{n,j} = \frac{e_n^\top e_j}{\|e_n\|\|e_j\|}, \quad m_j \in \mathcal{M}.$$

The top-k memories form the candidate set:

$$\mathcal{M}_{\text{near}}^n = \text{TopK}_{m_j \in \mathcal{M}}(s_{n,j}).$$

Then, the memory agent LM_{Mem} analyzes the new memory and its candidate memories to generate semantic links:

$$L_n \leftarrow \text{LM}_{\text{Mem}}\left(m_n \parallel \mathcal{M}_{\text{near}}^n \parallel P_{\text{link}}\right),$$

where P_{link} is the linking prompt. It guides LM_{Mem} to identify semantic relations between memory notes, including shared topics, complementary preferences, intent progression, preference-transition relations, and cross-domain transfer. The output L_n is a set of linked memory identifiers, optionally with relation descriptions, which is stored in the link field of m_n. In this way, collaborative relations become semantic and interpretable, rather than only numerical edge weights.

2.5 Dynamic Memory Evolution

A new interaction may change the interpretation of previous memories. Instead of only appending new memories, ATOMREC allows new evidence to trigger updates to related historical memories. For each candidate memory $m_j \in \mathcal{M}_{\text{near}}^n$, we decide whether to evolve it using a simple trigger condition:

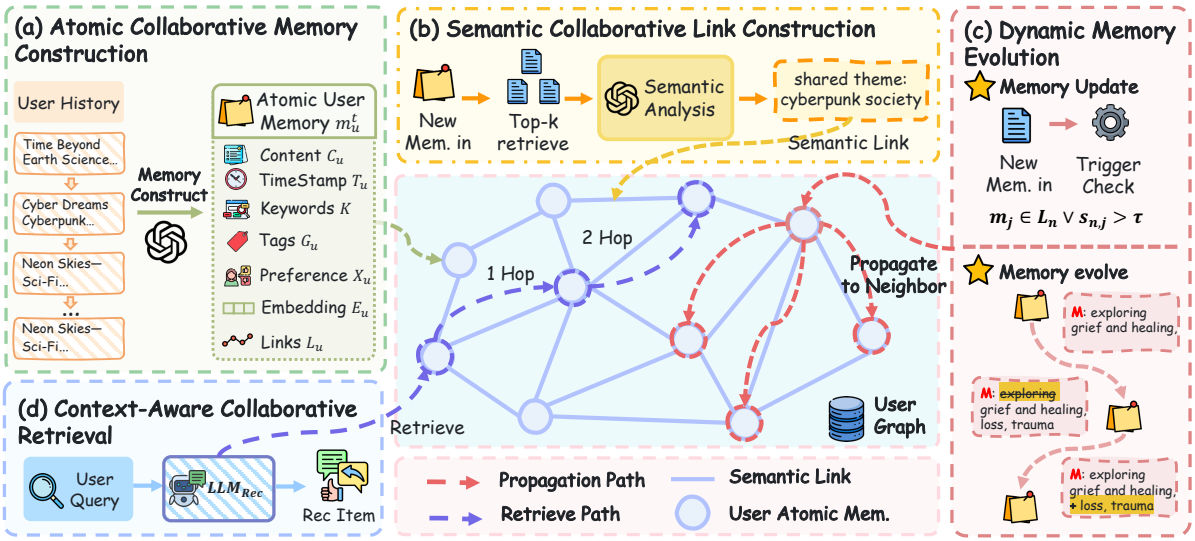

Figure 2: Architecture of AtomRec. Atomic memory construction, semantic collaborative link construction, dynamic memory evolution, and context-aware collaborative retrieval jointly support evidence-path recommendation.

$$\text{Evolve}(m_j) = I\left[m_j \in L_n \ \vee \ s_{n,j} > \tau_{\text{evo}}\right],$$

where L_n is the semantic link set of the new memory, $s_{n,j}$ is the similarity score between m_n and m_j, and τ_{evo} is the memory evolution threshold. If the trigger condition is satisfied, the memory agent updates the historical memory:

$$m_j^* \leftarrow \text{LM}_{\text{Mem}}\left(m_n \parallel m_j \parallel \mathcal{M}_{\text{near}}^n \parallel P_{\text{evolve}}\right),$$

where P_{evolve} instructs the agent to refine the memory while preserving its original semantics. The update is performed at the field level:

$$m_j^* = \{c_j^*, t_j, K_j^*, G_j^*, X_j^*, e_j^*, L_j^*\}.$$

Specifically, evolution may strengthen keywords, refine semantic tags, reconstruct the contextual description, and update semantic links. The timestamp t_j remains unchanged to preserve the original temporal position, while the semantic fields are revised to reflect newly observed evidence. This field-level evolution allows the memory graph to reorganize itself as user behavior changes.

2.6 Context-Aware Collaborative Retrieval and Recommendation

During recommendation, AtomRec retrieves a linked memory subgraph instead of isolated top-k neighbors. Given the current user state, we first retrieve an initial set of relevant memories:

$$\mathcal{M}_0^t = \text{TopK}_{m_i \in \mathcal{M}}\left(\cos(q_u^t, e_i)\right),$$

where q_u^t is the query representation derived from the current instruction, the user's recent interactions, and the pre-t atomic memory fields using the same encoder f_{enc}. We then expand the retrieved memories through semantic links:

$$\mathcal{G}_{\text{sub}}^t = \mathcal{M}_0^t \cup \bigcup_{m_i \in \mathcal{M}_0^t} L_i,$$

where the expansion can be extended to multiple hops in practice. We use relation descriptions as textual evidence for synthesis, while hop depth and retrieval similarity control subgraph expansion. The resulting subgraph contains not only similar memories, but also semantically linked evidence from related users and items.

The memory agent summarizes this subgraph into collaborative evidence:

$$M_{\text{collab}}^t \leftarrow \text{LM}_{\text{Mem}}\left(\mathcal{G}_{\text{sub}}^t \parallel P_{\text{synth}}\right),$$

where P_{synth} guides the agent to extract preference facets and evidence paths. Finally, the recommendation agent scores each candidate item $i \in \mathcal{C}_u^t$ based on the user's memory, collaborative evidence, and item memory:

$$r_{u,i}^t = \text{LM}_{\text{Rec}}\left(M_u^t \parallel M_{\text{collab}}^t \parallel M_i^t \parallel P_{\text{rank}}\right).$$

Candidate items are ranked by $r_{u,i}^t$. Because the retrieved context is built from semantic links

among atomic memories, the recommendation agent receives an evidence path rather than a set of disconnected neighbor summaries. This enables ATOMREC to combine semantic reasoning, collaborative enhancement, and memory evolution in a unified recommendation process.

3 Experimental Setup

We conduct experiments to answer the following research questions:

- **RQ1 (Overall Performance):** Does ATOMREC consistently improve recommendation accuracy over traditional, LM-based, agentic, and memory-augmented baselines?
- **RQ2 (Component Analysis):** How do atomic memory construction, semantic linking, memory evolution, link-aware retrieval, and key hyperparameters affect performance?
- **RQ3 (Preference Drift):** Does ATOMREC better capture users whose preferences shift over time?
- **RQ4 (Qualitative Behavior):** How does evolving atomic memory support interpretable evidence-path retrieval in concrete cases?
- **RQ5 (Efficiency):** What cost-performance trade-offs does ATOMREC exhibit under different methods and backbone models?

Datasets. We evaluate on four instruction-augmented benchmarks from MemRec (Chen et al., 2026): Amazon Books, Goodreads, MovieTV, and Yelp, originally adapted from InstructRec (Zhang et al., 2026b). We use the same preprocessed data and train/dev/test splits for fair comparison (Kang and McAuley, 2018; McAuley et al., 2015). Dataset statistics are provided in Appendix A.

Baselines. We compare ATOMREC with representative baselines from three categories: traditional recommenders, including LightGCN (He et al., 2020) and SASRec (Kang and McAuley, 2018); LM-based recommenders, including P5 (Geng et al., 2022) and Vanilla LLM (Liu et al., 2023); and agentic or memory-augmented recommenders, including iAgent (Xu et al., 2025), RecBot (Tsitseklis et al., 2023), AgentCF (Zhang et al., 2024b), i2Agent (Xu et al., 2025), and MemRec (Chen et al., 2026). Detailed baseline descriptions are provided in Appendix B.

Evaluation Protocols. We evaluate the main results on the full test sets with candidate size $N = 10$, reporting H@K and N@K for $K \in \{3, 5\}$. For ablation and analysis experiments, we use the same randomly sampled 1,000 users across methods following prior agentic recommendation settings (Xu et al., 2025; Chen et al., 2026).

Implementation Details. We use `gpt-4o-mini` (Hurst et al., 2024) as both the memory agent $\mathrm{LM}_{\mathrm{Mem}}$ and the recommendation agent $\mathrm{LM}_{\mathrm{Rec}}$ in the main experiments. We use frozen Sentence-T5 (Ni et al., 2022) as the text encoder f_{enc} to encode concatenated atomic fields for cosine-similarity retrieval. Unless otherwise specified, we set $k_{\mathrm{link}} = 20$, $\tau_{\mathrm{evo}} = 0.7$, and $h = 2$; these values are selected on the validation set and examined in the hyperparameter sensitivity analysis in Section 4.3. The memory agent constructs atomic memories, generates semantic links, evolves related historical memories, and synthesizes collaborative evidence, while the recommendation agent ranks candidate items based on the user instruction, user memory, collaborative evidence, and candidate item memories. During evaluation, memory states are built only from pre-target interactions, and the target item is used only for final reranking. More implementation details are provided in Appendix C.

4 Experimental Results

4.1 Performance (RQ1)

Table 1 presents the overall comparison across four datasets. ATOMREC achieves the best results on all datasets and all metrics, improving over the strongest baseline by around 8.5% across metrics.

- ATOMREC consistently outperforms all baselines on Books, Goodreads, MovieTV, and Yelp. The larger gains on Books and MovieTV suggest that atomic collaborative memory is especially useful for sparse and content-driven recommendation, showing **consistent cross-domain gains**.
- Agentic and memory-augmented baselines generally outperform traditional and LM-only methods, indicating that explicit semantic memory and collaborative evidence are important for instruction-aware ranking, confirming **the value of agentic memory**.
- Compared with MemRec, ATOMREC further benefits from structured fields, semantic links,

Table 1: Main results on four datasets. "Improv." denotes the relative improvement of our method over the best baseline. The best results are highlighted in bold, and the second-best results are underlined. All improvements over the strongest baseline are significant under paired bootstrap testing ($p < 0.05$).

Model	Books				Goodreads				MovieTV				Yelp			
	H@3	N@3	H@5	N@5	H@3	N@3	H@5	N@5	H@3	N@3	H@5	N@5	H@3	N@3	H@5	N@5
Traditional Recommenders																
LightGCN	0.3259	0.2596	0.5703	0.3592	0.5879	0.4432	0.7903	0.5263	0.5643	0.4738	0.6883	0.5241	0.5658	0.4720	0.7546	0.5494
SASRec	0.2830	0.2001	0.4845	0.2824	0.3518	0.2576	0.5407	0.3349	0.5233	0.4470	0.6382	0.4942	0.4312	0.3458	0.5597	0.3980
LM-based Recommenders																
P5	0.3607	0.2994	0.5273	0.3671	0.3229	0.2509	0.5060	0.3256	0.3206	0.2554	0.5008	0.3290	0.3207	0.2435	0.5220	0.4785
Vanilla LLM	0.5617	0.4533	0.7270	0.5226	0.4662	0.3948	0.7390	0.5041	0.7564	0.6098	0.8603	0.6445	0.5275	0.3696	0.6861	0.4360
Agentic Recommenders																
iAgent	0.5560	0.4858	0.6905	0.5409	0.4949	0.3954	0.6591	0.4626	0.6170	0.5361	0.7420	0.5871	0.6005	0.5148	0.7300	0.5681
RecBot	0.5491	0.4846	0.6786	0.5376	0.4754	0.3876	0.6495	0.4589	0.6113	0.5375	0.7309	0.5866	0.6003	0.5156	0.7169	0.5636
AgentCF	0.6060	0.4960	0.7403	0.5512	0.5910	0.4654	0.7726	0.5399	0.6693	0.5523	0.7864	0.6006	0.4374	0.3326	0.6374	0.4147
i2Agent	0.6517	0.5649	0.7708	0.6138	0.6079	0.4825	0.7675	0.5481	0.7225	0.6262	0.8221	0.6672	0.6454	0.5517	0.7648	0.6007
MemRec	0.6786	0.6078	0.7764	0.6480	0.6498	0.5490	0.7991	0.6005	0.7660	0.6907	0.8654	0.7068	0.6632	0.5914	0.7738	0.6251
Ours	**0.7462**	**0.6690**	**0.8543**	**0.7130**	**0.6980**	**0.5922**	**0.8558**	**0.6458**	**0.8323**	**0.7532**	**0.9364**	**0.7683**	**0.7119**	**0.6393**	**0.8330**	**0.6745**
Improv.	9.96%	10.07%	10.03%	10.03%	7.42%	7.86%	7.10%	7.55%	8.65%	9.05%	8.20%	8.70%	7.35%	8.10%	7.65%	7.90%

and field-level evolution. These designs enable connected evidence-path retrieval rather than isolated neighbor summaries, supporting **fine-grained atomic evidence**.

4.2 Ablation Study (RQ2)

Table 2 reports the ablation results on Books, with additional datasets in Appendix E. Removing any component hurts performance, confirming that AtomRec's gains come from the joint design of atomic memory, semantic linking, memory evolution, and link-aware retrieval.

- **Atomic fields matter.** Collapsing atomic notes into coarse summaries causes the largest drop, while removing keywords/tags or context also hurts performance.
- **Semantic links help.** Removing links or replacing them with embedding-only neighbors reduces performance, showing the value of relation-aware evidence.
- **Evolution and path retrieval help.** Append-only memory and removing link-aware retrieval both underperform, showing that evolving memories and connected evidence paths improve ranking.

4.3 Hyperparameter Sensitivity (RQ2)

Figure 3 shows the sensitivity of four memory-side hyperparameters on Books using H@5 and N@5. AtomRec performs best when $k_{\text{link}} = 20$, $\tau_{\text{evo}} = 0.7$, $h = 2$, and $k_{\text{ret}} = 15$. These results suggest that the memory mechanism benefits from moderate semantic linking, conservative memory evolution, limited multi-hop retrieval, and a compact retrieval candidate set. Smaller values may miss useful collaborative evidence, while larger values can introduce noisy links or weakly related memories.

Table 2: Ablation study of AtomRec on the Books dataset. Avg. Δ denotes the average relative performance decrease across all four metrics compared with the full model.

Variant	H@3	N@3	H@5	N@5	Δ_{avg}
Full AtomRec	**0.7462**	**0.6690**	**0.8543**	**0.7130**	–
w/o Atomic Mem.	0.7046	0.6308	0.8109	0.6742	5.45%
w/o Keywords/Tags	0.7159	0.6416	0.8238	0.6854	3.90%
w/o Context Field	0.7227	0.6480	0.8320	0.6927	2.94%
w/o Collab. Link.	0.7168	0.6424	0.8255	0.6860	3.77%
Embedding-only Links	0.7235	0.6496	0.8334	0.6936	2.78%
Append-only Memory	0.7251	0.6512	0.8350	0.6951	2.56%
w/o Link Retrieval	0.7292	0.6567	0.8391	0.7004	1.92%

4.4 Preference Drift Analysis (RQ3)

We examine whether AtomRec better handles users with changing interests. For each user, we split the history into early and late segments and compute the drift score as $d_u = 1 - \cos(\bar{e}_u^{\text{early}}, \bar{e}_u^{\text{late}})$, where $\bar{e}_u^{\text{early}}$ and $\bar{e}_u^{\text{late}}$ are the averaged item embeddings of the two segments. Users are then divided into Low, Medium, and High Drift groups. As shown in Figure 4, all methods degrade as preference drift increases, while AtomRec obtains larger gains over MemRec on higher-drift users. This suggests that field-level memory evolution and semantic evidence paths

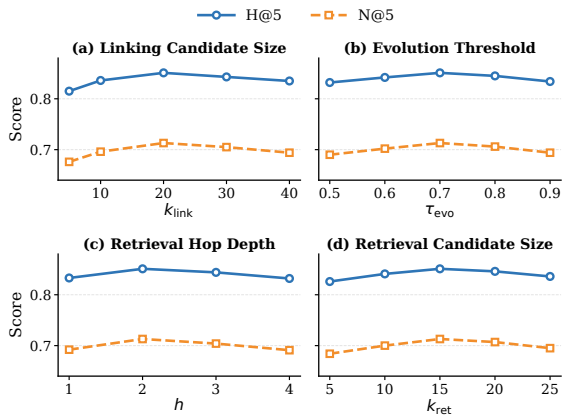

Figure 3: Hyperparameter sensitivity of AtomRec on Books. We report H@5 and N@5 while varying the linking candidate size k_{link}, memory evolution threshold τ_{evo}, retrieval hop depth h, and retrieval candidate size k_{ret}.

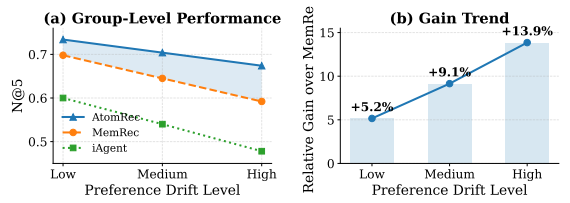

Figure 4: Preference drift analysis. (a) N@5 performance across user groups with different drift levels. (b) Relative gain of AtomRec over MemRec increases as preference drift becomes stronger.

help preserve emerging interests under preference shifts.

4.5 Case Study (RQ4)

Figure 5 shows a representative case where the user's history shifts from spiritual communication, to grief healing, and then to etheric and afterlife-evidence books. Coarse memory captures these broad themes but repeatedly rewrites them into one summary, mixing preference stages and weakening stage-level evidence. In contrast, AtomRec preserves atomic fields and performs field-level edits, keeping earlier evidence accessible while refining current intent. The retrieval comparison further shows that the target item, *On the Edge of the Etheric*, is missed without atomic memory and only partially matched without link-aware retrieval, while full AtomRec follows 1-hop and 2-hop evidence links to rank the target item first.

4.6 Efficiency and Deployment Analysis (RQ5)

We analyze deployment cost using cost-performance trends and token/cost statistics, with details in Appendix J. Figures 6 and 7 show two findings:

- **Cost-performance trade-off.** AtomRec achieves higher N@5 on Books than representative LLM-based and agentic baselines with moderate extra cost. Stronger backbones bring only modest additional gains, suggesting that the improvement mainly comes from the memory mechanism rather than backbone scaling alone.
- **Online reranking overhead.** AtomRec uses more tokens than MemRec due to atomic construction, semantic linking, field-level evolution, and evidence synthesis. Since these memory-side operations can be cached and executed asynchronously, the additional online reranking overhead remains limited.

5 Related Work

Memory in LLM Agents. While LLMs excel in long-horizon reasoning (Achiam et al., 2023; Ouyang et al., 2022; Brown et al., 2020; Yao et al., 2022), their restricted context windows necessitate external memory for persistent knowledge retention (Hu et al., 2025; Zhang et al., 2025b; Du, 2026; Park et al., 2023; Shinn et al., 2023). Early architectures rely on predefined workflows, like MemoryBank's forgetting curves (Zhong et al., 2024), MemGPT's hierarchical buffers (Packer et al., 2023), and SCM's read-write streams (Wang et al., 2023). Later models (e.g., A-Mem (Xu et al., 2026), Mem0 (Chhikara et al., 2025), MemInsight (Salama et al., 2025)) enhance adaptability via associative linking and layered summarization (Fang et al., 2025; Li et al., 2025). However, they predominantly focus on single-agent memory, leaving collaborative memory across users and items underexplored for interactive recommendation.

LLM-based Recommendation Agents. LLM agents support recommendation through simulation, planning, tool use, and memory-driven personalization (Peng et al., 2025; Zhang et al., 2025a; Zhu et al., 2025). Agent4Rec (Zhang et al., 2024a) and AgentCF (Zhang et al., 2024b) simulate user–item dynamics, while RecMind (Wang

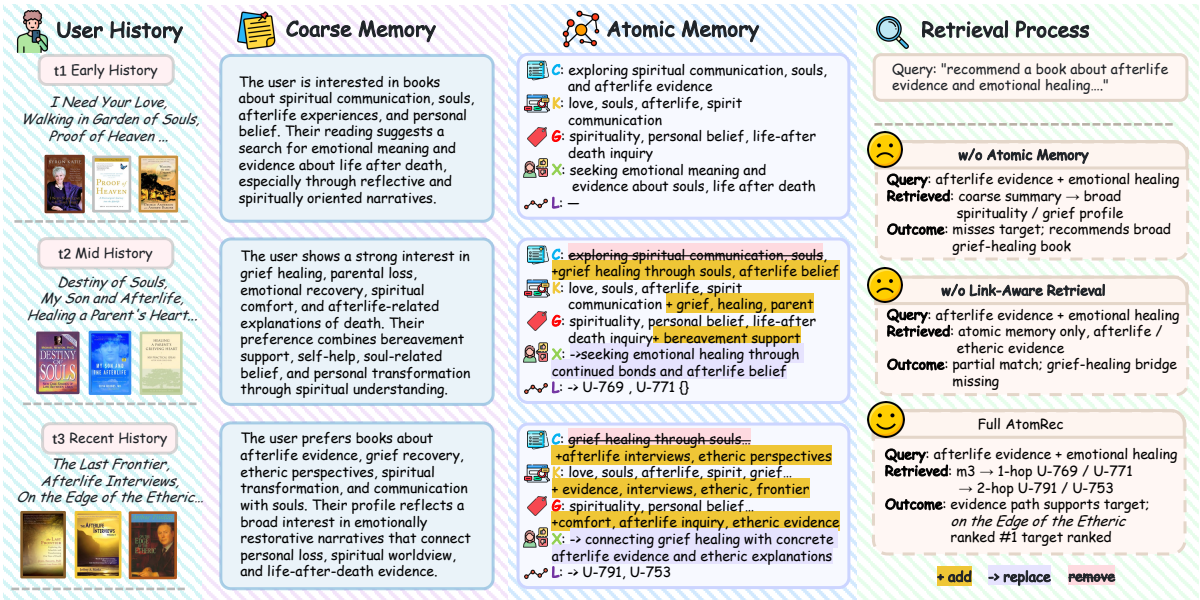

Figure 5: Case study of coarse memory rewriting and atomic memory evolution on Amazon Books. AtomRec preserves structured atomic fields and uses link-aware retrieval to recover the target item.

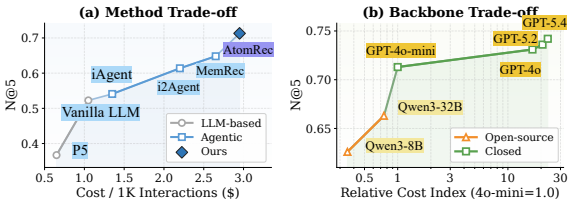

Figure 6: Efficiency-aware analysis of AtomRec on Books. (a) Method-level cost-performance trade-off. (b) Backbone-level cost-performance trade-off, with cost normalized by gpt-4o-mini.

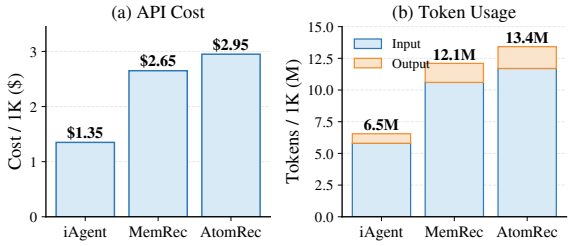

Figure 7: Absolute efficiency breakdown under the default lightweight backbone. We report estimated API cost and token usage per 1K interactions.

et al., 2024a), InteRecAgent (Huang et al., 2025), and MACRec (Wang et al., 2024b) explore task-oriented agent workflows. Recent systems further improve adaptive reasoning: ChainRec (Li et al., 2026b) routes standardized tools, RecThinker (Zhang et al., 2026a) plans tool calls through information-gap analysis, and RecNet (Li et al., 2026a) propagates preference updates through router agents. For long-term personalization, iAgent (Xu et al., 2025) and RecBot (Tsitsiklis et al., 2023) update isolated user profiles, while MemRec (Chen et al., 2026) builds a collaborative memory graph. In contrast, AtomRec focuses on the memory substrate itself by decomposing coarse memories into field-structured atomic notes, linking them semantically, and evolving related historical memories for evidence-path recommendation.

6 Conclusion

We present AtomRec, an agentic recommender system with evolving atomic collaborative memory. AtomRec represents user and item memories as structured atomic notes, connects them through semantic links, and evolves related historical memories when new interactions arrive. By replacing coarse memory summaries and scalar collaborative edges with atomic memory units and semantic evidence paths, AtomRec supports fine-grained long-term preference modeling. Experiments on four instruction-augmented benchmarks show consistent improvements over traditional, LLM-based, agentic, and memory-augmented baselines, while ablation studies and qualitative analysis confirm the benefits of atomic construction, semantic collaborative link construction, and dynamic evolution. These results suggest that making memory more structured, linkable, and evolvable is a promising direction for building more interpretable and adaptive recommender agents.

Limitations

ATOMREC has three main limitations. First, memory evolution may over-compress user interests into a dominant subtheme, causing semantically adjacent false positives when fine-grained candidate discrimination is required. Our qualitative analysis in Appendix L shows that the model can capture the correct broad preference region but still under-rank the target item when nearby candidates share partial emotional or spiritual cues. Second, the model may over-emphasize stable long-term preference trajectories and under-rank idiosyncratic or short-term exploratory targets. Third, collaborative signals may be absorbed into memory fields without being preserved as explicit final-step links, which weakens post-hoc traceability. Future work should improve fine-grained reranking, uncertainty-aware modeling of exploratory behavior, and provenance preservation for semantic memory links.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, pages 1007–1014.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Weixin Chen, Yuhan Zhao, Jingyuan Huang, Zihe Ye, Clark Mingxuan Ju, Tong Zhao, Neil Shah, Li Chen, and Yongfeng Zhang. 2026. Memrec: Collaborative memory-augmented agentic recommender system. *arXiv preprint arXiv:2601.08816*.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

Aminu Da'u and Naomie Salim. 2020. Recommendation system based on deep learning methods: a systematic review and new directions. *Artificial Intelligence Review*, 53(4):2709–2748.

Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasirzadeh, and Silvia Milano. 2024. A review of modern recommender systems using generative models (gen-recsys). In *Proceedings of the 30th ACM SIGKDD conference on Knowledge Discovery and Data Mining*, pages 6448–6458.

Pengfei Du. 2026. Memory for autonomous llm agents: Mechanisms, evaluation, and emerging frontiers. *arXiv preprint arXiv:2603.07670*.

Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, and 1 others. 2025. A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems. *arXiv preprint arXiv:2508.07407*.

Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chatrec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems*, pages 299–315.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European conference on information retrieval*, pages 364–381. Springer.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, and 1 others. 2025. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*.

Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2025. Recommender ai agent: Integrating large language models for interactive recommendations. *ACM Transactions on Information Systems*, 43(4):1–33.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Clark Mingxuan Ju, Liam Collins, Leonardo Neves, Bhuvesh Kumar, Louis Yufeng Wang, Tong Zhao,

and Neil Shah. 2025. Generative recommendation with semantic ids: A practitioner's handbook. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 6420–6425.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Hyeyoung Ko, Suyeon Lee, Yoonseo Park, and Anna Choi. 2022. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics*, 11(1):141.

Xiaoyu Kong, Leheng Sheng, Junfei Tan, Yuxin Chen, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. 2025. Minionerec: An open-source framework for scaling generative recommendation. *arXiv preprint arXiv:2510.24431*.

Bingqian Li, Xiaolei Wang, Junyi Li, Weitao Li, Long Zhang, Sheng Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2026a. Recnet: Self-evolving preference propagation for agentic recommender systems. *arXiv preprint arXiv:2601.21609*.

Fuchun Li, Qian Li, Xingyu Gao, Bocheng Pan, Yang Wu, Jun Zhang, Huan Yu, Jie Jiang, Jinsheng Xiao, and Hailong Shi. 2026b. Chainrec: An agentic recommender learning to route tool chains for diverse and evolving interests. *arXiv preprint arXiv:2602.10490*.

Xiaopeng Li, Pengyue Jia, Derong Xu, Yi Wen, Yingyi Zhang, Wenlin Zhang, Wanyu Wang, Yichao Wang, Zhaocheng Du, Xiangyang Li, and 1 others. 2025. A survey of personalization: From rag to agent. *ACM Transactions on Information Systems*.

Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*.

Qidong Liu, Xiangyu Zhao, Yuhao Wang, Yejing Wang, Zijian Zhang, Yuqi Sun, Xiang Li, Maolin Wang, Pengyue Jia, Chong Chen, and 1 others. 2024. Large language model enhanced recommender systems: A survey. *arXiv preprint arXiv:2412.13432*.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pages 1864–1874.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. 2023. Memgpt: towards llms as operating systems.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Qiyao Peng, Hongtao Liu, Hua Huang, Qing Yang, and Minglai Shao. 2025. A survey on llm-powered agents for recommender systems. *arXiv preprint arXiv:2502.10050*.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, and 1 others. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. Meminsight: Autonomous memory augmentation for llm agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33124–33140.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Konstantinos Tsitseklis, Georgia Stavropoulou, Anastasios Zafeiropoulos, Athina Thanou, and Symeon Papavassiliou. 2023. Recbot: Virtual museum navigation through a chatbot assistant and personalized recommendations. In *Adjunct proceedings of the 31st ACM conference on user modeling, adaptation and personalization*, pages 388–396.

Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM conference on recommender systems*, pages 86–94.

Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. 2023. Enhancing large language model with self-controlled memory framework. *arXiv preprint arXiv:2304.13343*.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024a. Recmind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4351–4364.

Zhefan Wang, Yuanqing Yu, Wendi Zheng, Weizhi Ma, and Min Zhang. 2024b. Macrec: A multi-agent collaboration framework for recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2760–2764.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, and 1 others. 2024. A survey on large language models for recommendation. *World Wide Web*, 27(5):60.

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

Wujiang Xu, Yunxiao Shi, Zujie Liang, Xuying Ning, Kai Mei, Kun Wang, Xi Zhu, Min Xu, and Yongfeng Zhang. 2025. iagent: Llm agent as a shield between user and recommender systems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18056–18084.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Runyang You, Yongqi Li, Xinyu Lin, Xin Zhang, Wenjie Wang, Wenjie Li, and Liqiang Nie. 2026. R^{2}ec: Towards large recommender models with reasoning. *Advances in Neural Information Processing Systems*, 38:62376–62405.

Weiqi Yue, Yuyu Yin, Xin Zhang, Binbin Shi, Tingting Liang, and Jian Wan. 2025. Cot4rec: Revealing user preferences through chain of thought for recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13142–13151.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024a. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, pages 1807–1817.

Haobo Zhang, Yutao Zhu, Kelong Mao, Tianhao Li, and Zhicheng Dou. 2026a. Recthinker: An agentic framework for tool-augmented reasoning in recommendation. *arXiv preprint arXiv:2603.09843*.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024b. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pages 3679–3689.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2026b. Recommendation as instruction following: A large language model empowered recommendation approach. *ACM Transactions on Information Systems*, 43(5):1–37.

Yu Zhang, Shutong Qiao, Jiaqi Zhang, Tzu-Heng Lin, Chen Gao, and Yong Li. 2025a. A survey of large language model empowered agents for recommendation and search: Towards next-generation information retrieval. *arXiv preprint arXiv:2503.05659*.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025b. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1435–1448. IEEE.

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

Xi Zhu, Yu Wang, Hang Gao, Wujiang Xu, Chen Wang, Zhiwei Liu, Kun Wang, Mingyu Jin, Linsey Pang, Qingsong Weng, and 1 others. 2025. Recommender systems meet large language model agents: A survey. *Foundations and Trends® in Privacy and Security*, 7(4):247–396.

A Dataset Details

We used the same preprocessed representation and train/dev/test splits as MemRec (Chen et al., 2026) for all datasets. The four benchmarks were originally adapted from InstructRec (Zhang et al., 2026b) and cover different recommendation domains, including book recommendation, social reading, movies and TV shows, and local business services. Table 3 reports the dataset statistics used in our experiments.

Books. (McAuley et al., 2015; Ni et al., 2019) is a large-scale book recommendation benchmark derived from the Amazon review corpus. It contains the largest item space among the four datasets and exhibits highly sparse user–item interactions. User preferences in this domain are often content-driven and relatively stable, involving genres, authors, writing styles, and recurring themes. This makes Books a challenging setting for modeling long-tail interests and sparse collaborative signals.

Goodreads. (Wan and McAuley, 2018) is collected from a social book cataloging platform and has the densest interaction structure in our evaluation. Compared with Books, users in Goodreads usually have longer interaction histories and stronger reading continuity. The dataset also reflects community-driven reading behavior, where users may repeatedly engage with book series, authors, or socially popular titles. This setting is useful for evaluating whether memory can capture persistent and evolving reading preferences.

MovieTV. (McAuley et al., 2015; Ni et al., 2019) covers movies and TV shows from the Amazon review corpus. Unlike book-centered domains, user interests in this dataset can be more dynamic because viewing choices are often affected by genre, cast, release recency, and short-term entertainment context. The relatively shorter average sequence length also makes it harder to infer stable preferences, requiring the model to balance long-term taste with recent behavioral signals.

Yelp. (Asghar, 2016) contains local business recommendations, such as restaurants and services. This domain differs from media recommendation because user decisions are strongly constrained by category, location, price range, and situational intent. Preferences are therefore more context-dependent, and similar historical behaviors may not always imply the same future choice. Yelp provides a useful testbed for evaluating memory mechanisms under local-service and attribute-sensitive recommendation scenarios.

Dataset	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{E}\rvert$	$\bar{L}_u$	Density
Books	7.4K	120.9K	207.8K	28.2	2.33e-4
Goodreads	11.7K	57.4K	618.3K	52.7	9.19e-4
MovieTV	5.6K	29.0K	79.7K	14.1	4.87e-4
Yelp	3.0K	31.6K	63.1K	21.4	6.77e-4

Table 3: Statistics of the datasets used in our experiments.

B Detailed Baseline Descriptions

We provide detailed descriptions of the baselines used in our experiments. These methods cover traditional recommenders, LM-based recommenders, agentic recommenders, and memory-augmented agentic recommenders.

Traditional Recommenders.

- **LightGCN** (He et al., 2020) is a graph-based collaborative filtering model that simplifies graph convolution by removing feature transformation and nonlinear activation. It learns user and item embeddings through neighborhood aggregation on the user–item interaction graph.
- **SASRec** (Kang and McAuley, 2018) is a self-attentive sequential recommender that models user interaction histories with a Transformer-style attention mechanism and predicts the next item from sequential preference patterns.

LM-Based Recommenders.

- **P5** (Geng et al., 2022) formulates recommendation as a text-to-text language modeling problem through personalized prompts, allowing a pretrained language model to handle multiple recommendation tasks in a unified format.
- **Vanilla LLM** (Liu et al., 2023) directly prompts a large language model to rank candidate items from user history and item descriptions. It does not maintain external memory or construct a collaborative memory graph.

Agentic Recommenders.

- **iAgent** (Xu et al., 2025) uses an LLM agent as an intermediary between the user and recommender system. It relies mainly on user-side instructions and profile information, making it a representative static-memory agentic baseline.

- **RecBot** (Tsitseklis et al., 2023) maintains and updates user preference memory during recommendation, but its memory is mainly organized around individual user states rather than an atomic collaborative memory graph.
- **AgentCF** (Zhang et al., 2024b) treats users and items as autonomous language agents and simulates user–item interactions for collaborative filtering. It captures collaborative behavior through agent interaction, but does not organize memory as fine-grained atomic notes with semantic links.

Memory-Augmented Agentic Recommender.

- **MemRec** (Chen et al., 2026) is the closest baseline to our work. It decouples memory management from recommendation reasoning and maintains a dynamic collaborative memory graph for downstream recommendation. In contrast, AtomRec focuses on the granularity and structure of memory itself: it represents user and item memories as atomic semantic notes, builds semantic links among them, and evolves related historical memories at the field level.

C Implementation Details

We provide additional implementation details for AtomRec, including backbone agents, memory writing, retrieval, output parsing, and evaluation protocol.

Backbone Agents. We use `gpt-4o-mini` (Hurst et al., 2024) as the default backbone for both the memory agent $\mathrm{LM}_{\mathrm{Mem}}$ and the recommendation agent $\mathrm{LM}_{\mathrm{Rec}}$. The memory agent performs atomic memory construction, semantic linking, memory evolution, and evidence synthesis. The recommendation agent ranks candidate items using the user instruction, user memory, synthesized collaborative evidence, and candidate item memories. Unless otherwise specified, all LLM-based methods use the same backbone and deterministic decoding setting for fair comparison. In the efficiency analysis, we additionally compare open-source Qwen3 backbones and closed GPT-family backbones.

Atomic Memory Writing. For each newly observed pre-target interaction, AtomRec creates a user-side atomic note that summarizes the interaction evidence from the user's perspective. If the interacted item does not already have an item memory, we also create an item-side atomic note from its available metadata and interaction context. Thus, each interaction creates at most one new user note and one new item note. User and item notes are stored in the same memory space $\mathcal{M}$ and can be connected through semantic links. Existing notes are not overwritten during construction; they are revised only through the dynamic memory evolution step.

Each atomic note has the form

$$m = \{c, t, K, G, X, e, L\},$$

where c is the textual memory content, t is the timestamp, K is the keyword set, G is the semantic tag set, X is the contextual description, e is the dense embedding, and L stores linked memory identifiers. We use frozen Sentence-T5 (Ni et al., 2022) as the text encoder f_{enc}. The embedding e is computed from the concatenation of textual fields (c, K, G, X) and ℓ_2-normalized before cosine-similarity retrieval.

Semantic Collaborative Link Construction. When a new memory note m_n is inserted, we first retrieve the top-k_{link} nearest notes from $\mathcal{M}$ according to cosine similarity. We set $k_{\mathrm{link}} = 20$ by default. The memory agent then analyzes m_n and the retrieved candidates to generate semantic links. The output is a structured JSON object containing linked note identifiers and optional relation descriptions. These links form the atomic collaborative memory graph and support later link-aware retrieval.

Dynamic Memory Evolution. For each nearby historical note m_j, AtomRec triggers field-level evolution if m_j is linked to the new note or if its similarity score exceeds the evolution threshold:

$$m_j \in L_n \ \vee \ s_{n,j} > \tau_{\mathrm{evo}}.$$

We set $\tau_{\mathrm{evo}} = 0.7$ by default. When evolution is triggered, the memory agent may update the keywords, tags, contextual description, and links of the historical note, while preserving its original timestamp. The updated embedding is recomputed from the revised fields. If the memory agent outputs an invalid or unsupported update, we keep the previous note unchanged.

Context-Aware Collaborative Retrieval. During recommendation, AtomRec retrieves relevant notes from the pre-target memory state using the query representation derived from the user instruction and pre-t atomic memory fields. It then expands the retrieved set through semantic links up

to hop depth h, where $h = 2$ by default. Relation descriptions are used as textual evidence during synthesis rather than as learned edge weights. The memory agent compresses the retrieved subgraph into collaborative evidence, which is then passed to the recommendation agent.

Candidate Ranking and Score Parsing. For each test instance, the recommendation agent receives the user instruction, user memory, synthesized collaborative evidence, candidate item memories, and the candidate item set. The agent is instructed to return a structured JSON ranking with item-level relevance scores. We parse these scores as $r^t_{u,i}$ and sort candidate items accordingly. If the response contains a valid ranked list but no explicit numeric scores, we convert the returned order into ranking scores for evaluation. If the output is not parseable, we apply one retry with a stricter JSON-only instruction; if the retry still fails, we discard the invalid response.

Hyperparameters. The default hyperparameters are selected according to validation performance and used across datasets unless otherwise specified:

$$k_{\text{link}} = 20, \quad \tau_{\text{evo}} = 0.7, \quad h = 2.$$

We analyze their sensitivity in Section 4.3. The results show that moderate linking size, conservative evolution threshold, and limited multi-hop retrieval provide the best overall performance.

Evaluation Protocol. The main comparison is conducted on the full test sets with candidate size $N = 10$. For ablation studies, hyperparameter sensitivity, preference drift analysis, and semantic-link analysis, we use the same sampled subset across compared methods. Efficiency statistics are computed from logged LLM calls under the default lightweight backbone and reported per 1K interactions.

During evaluation, AtomRec uses a strict temporal memory snapshot. For each test instance at time t, atomic construction, semantic linking, memory evolution, and retrieval use only interactions before t. The held-out target item is visible only as one candidate during final reranking and is never used to construct, link, or evolve memory for the same instance. A detailed leakage-control checklist is provided in Appendix G.

LLM Decoding and Output Parsing. For all LLM calls, we use deterministic decoding with temperature set to 0.0. The memory agent and recommendation agent are instructed to return structured JSON outputs for atomic fields, semantic links, evolution decisions, evidence summaries, and item scores. We parse outputs with a rule-based JSON parser. If a response is malformed, we apply one retry with the same input and an additional formatting instruction. If the retry still fails, we keep the previous memory state unchanged or discard the invalid ranking output. This conservative fallback prevents malformed outputs from introducing uncontrolled memory changes.

D Additional Experimental Results

To examine whether the advantage of AtomRec remains under a more challenging reranking setting, we further evaluate all methods with a larger candidate set size of $N = 20$. Tables 4 and 5 report the results on four datasets. Compared with the default $N = 10$ setting, this evaluation increases the number of distractor candidates and therefore provides a stricter test of ranking robustness. AtomRec consistently outperforms MemRec on most metrics, showing that atomic memory construction, semantic linking, and memory evolution remain effective when the candidate set becomes larger. Unless otherwise specified, all positive improvements over the strongest baseline are significant under paired bootstrap testing ($p < 0.05$).

E Additional Ablation Results

We provide additional ablation results on Books and Goodreads to further examine the contribution of both module-level and mechanism-level designs in AtomRec. We consider variants that remove atomic memory fields, remove specific atomic fields, replace semantic links with embedding-only links, disable memory evolution, and remove link-aware retrieval. Avg. Δ denotes the average relative performance decrease across H@3, N@3, H@5, and N@5 compared with the full model on each dataset.

Table 6 shows consistent trends on both Books and Goodreads. Removing the full atomic memory representation causes the largest drop, confirming that field-structured notes are more effective than coarse textual summaries. The field-level variants further show that keywords/tags and contextual descriptions both contribute to performance. Replacing LLM-guided semantic links with embedding-only links also reduces performance, indicating that the linking module provides relation-aware evidence beyond nearest-neighbor similarity. Finally,

Table 4: Main results on Books and Goodreads with a larger candidate set ($N = 20$). "Improv." denotes the relative improvement of our method over the best baseline. The best results are highlighted in bold, and the second-best results are underlined.

Model	Books					Goodreads				
	H@1	H@5	N@5	H@10	N@10	H@1	H@5	N@5	H@10	N@10
Traditional Recommenders										
LightGCN	0.1276	0.2622	0.1947	0.5512	0.2854	0.1617	0.5566	0.3588	0.8177	0.4434
SASRec	0.0453	0.2353	0.1378	0.4896	0.2188	0.0699	0.3053	0.1859	0.5435	0.2621
LM-based Recommenders										
P5	0.1648	0.3051	0.2331	0.5216	0.3022	0.1038	0.2611	0.1798	0.5041	0.2572
Vanilla LLM	0.1730	0.4155	0.2955	0.6129	0.3599	0.0999	0.3245	0.2211	0.6712	0.3291
Agentic Recommenders										
iAgent	0.3258	0.5069	0.4173	0.6209	0.4537	0.1621	0.4107	0.2871	0.6035	0.3490
RecBot	0.2471	0.4030	0.3247	0.5768	0.3801	0.1234	0.3364	0.2289	0.5583	0.2999
AgentCF	0.2470	0.5481	0.4026	0.7250	0.4594	0.1875	0.5427	0.3692	0.7805	0.4462
i2Agent	0.3712	0.5947	0.4874	0.7387	0.5336	0.2065	0.5350	0.3767	0.7428	0.4435
MemRec	0.4236	0.6351	0.5332	0.7667	0.5756	0.2657	0.6062	0.4434	0.7948	0.5042
Ours	**0.4533**	**0.6700**	**0.5679**	**0.7958**	**0.6090**	**0.2795**	**0.6347**	**0.4691**	**0.8250**	**0.5339**
Improv.	7.01%	5.50%	6.51%	3.80%	5.80%	5.19%	4.70%	5.80%	0.89%	5.89%

append-only memory and removing link-aware retrieval both underperform the full model, supporting the value of field-level memory evolution and connected evidence-path retrieval.

F Additional Diagnostic Analyses

Cost-Matched Control. To examine whether the improvement of ATOMREC comes only from using more LLM context, we conduct a cost-matched control on the Books analysis subset. We restrict the retrieved evidence budget of ATOMREC so that its average token usage is close to MemRec under the same backbone and decoding setting. This variant keeps atomic memory construction, semantic linking, and field-level evolution, but limits the number of retrieved notes and synthesized evidence tokens during final reranking.

Table 7 shows that cost-matched ATOMREC still outperforms MemRec by about 6.5% on average under a similar token budget, although the margin is smaller than the full setting. This suggests that the improvement does not only come from longer LLM context; the atomic memory structure and semantic evidence paths also contribute to ranking performance.

Evolution Audit. We further inspect whether field-level memory evolution preserves historical semantics while incorporating new evidence. We sample evolved notes from the Books analysis subset and evaluate each update along three dimensions: *Semantic Preservation*, which measures whether the updated note remains consistent with the original memory; *Evidence Support*, which measures whether the new fields are supported by the trigger note or nearby linked notes; and *Recommendation Usefulness*, which measures whether the update provides clearer evidence for downstream ranking. Each dimension is rated on a 1–5 scale by an automatic judge using a prompt different from the memory agent.

Table 8 suggests that most evolved notes preserve the original memory semantics and are supported by nearby evidence. The lower usefulness score indicates that not every update directly benefits ranking, which is consistent with the failure cases in Appendix L. In qualitative inspection, harmful cases mainly arise when semantically adjacent themes are over-merged or when a dominant long-term preference absorbs a short-term exploratory signal. These observations motivate future work on edit provenance, confidence-aware evolution, and rollback mechanisms.

G Temporal Evaluation Protocol and Leakage Control

We use a strictly chronological evaluation protocol to prevent temporal leakage. For each test instance $(u, \mathcal{H}_u^{t-1}, i_t, \mathcal{C}_u)$, ATOMREC constructs and updates the memory state using only interactions before time t. Atomic memory construction, semantic linking, memory evolution, and collaborative evidence synthesis never access the held-out

Table 5: Main results on MovieTV and Yelp with a larger candidate set ($N = 20$). "Improv." denotes the relative improvement of our method over the best baseline. The best results are highlighted in bold, and the second-best results are underlined.

Model	MovieTV					Yelp				
	H@1	H@5	N@5	H@10	N@10	H@1	H@5	N@5	H@10	N@10
Traditional Recommenders										
LightGCN	0.2657	0.5330	0.4064	0.6815	0.4537	0.2549	0.5437	0.4046	0.7481	0.4692
SASRec	0.2923	0.5128	0.4092	0.6311	0.4470	0.1678	0.3993	0.2879	0.5590	0.3389
LM-based Recommenders										
P5	0.1113	0.2769	0.1902	0.5137	0.2657	0.0634	0.2492	0.1537	0.5051	0.2354
Vanilla LLM	0.2379	0.5003	0.3648	0.7261	0.4406	0.0254	0.1461	0.0831	0.5128	0.2010
Agentic Recommenders										
iAgent	0.3236	0.5362	0.4331	0.6762	0.4778	0.3236	0.5658	0.4499	0.6597	0.4799
RecBot	0.2420	0.4201	0.3316	0.6015	0.3895	0.1949	0.3742	0.2851	0.5519	0.3414
AgentCF	0.2870	0.6288	0.4648	0.7616	0.5077	0.1115	0.3897	0.2512	0.6372	0.3309
i2Agent	0.3822	0.6367	0.5178	0.7735	0.5617	0.3287	0.6083	0.4744	0.7562	0.5216
MemRec	0.4750	0.7543	0.6212	0.8752	0.6606	0.3620	0.6329	0.5035	0.7708	0.5478
Ours	**0.5083**	**0.7996**	**0.6603**	**0.9067**	**0.6976**	**0.3845**	**0.6652**	**0.5342**	**0.7917**	**0.5785**
Improv.	7.01%	6.00%	6.30%	3.60%	5.60%	6.22%	5.10%	6.10%	2.71%	5.60%

Table 6: Additional ablation results of ATOMREC on Books and Goodreads. Avg. Δ denotes the average relative performance decrease across all four metrics compared with the full model on each dataset.

Variant	Books					Goodreads				
	H@3	N@3	H@5	N@5	Avg. Δ	H@3	N@3	H@5	N@5	Avg. Δ
Full ATOMREC	**0.7462**	**0.6690**	**0.8543**	**0.7130**	–	**0.6980**	**0.5922**	**0.8558**	**0.6458**	–
w/o Atomic Memory	0.7046	0.6308	0.8109	0.6742	5.45%	0.6639	0.5585	0.8169	0.6080	5.24%
w/o Keywords/Tags	0.7159	0.6416	0.8238	0.6854	3.90%	0.6726	0.5678	0.8264	0.6177	3.89%
w/o Context Field	0.7227	0.6480	0.8320	0.6927	2.94%	0.6818	0.5769	0.8380	0.6277	2.45%
w/o Collaborative Linking	0.7168	0.6424	0.8255	0.6860	3.77%	0.6740	0.5684	0.8279	0.6188	3.72%
Embedding-only Links	0.7235	0.6496	0.8334	0.6936	2.78%	0.6836	0.5781	0.8400	0.6296	2.20%
Append-only Memory	0.7251	0.6512	0.8350	0.6951	2.56%	0.6850	0.5796	0.8412	0.6310	2.00%
w/o Link-Aware Retrieval	0.7292	0.6567	0.8391	0.7004	1.92%	0.6901	0.5847	0.8480	0.6372	1.16%

Table 7: Cost-matched control on the Books analysis subset. Avg. Tok. denotes average token usage per interaction.

Method	Avg. Tok.	H@3	N@3	H@5	N@5
MemRec	12.1K	0.6786	0.6078	0.7764	0.6480
ATOMREC (cost-matched)	12.3K	0.7224	0.6483	0.8267	0.6906
ATOMREC (full)	13.4K	**0.7462**	**0.6690**	**0.8543**	**0.7130**

Table 8: Evolution audit on sampled evolved notes from the Books analysis subset. Scores are rated on a 1–5 scale. Higher is better.

Update Type	Preservation	Support	Usefulness
Field-level Evolution	4.27	4.05	3.91

target interaction or any post-t interaction. The target item i_t is used only as one candidate in $\mathcal{C}_u$ during the final reranking stage by $\mathrm{LM}_{\mathrm{Rec}}$, where all candidate items are treated symmetrically. After ranking, the target interaction is not written back into memory for evaluating the same instance.

This protocol is especially important for ATOMREC, because historical atomic notes may be evolved when new evidence arrives. During evaluation, memory evolution is triggered only by pre-target interactions. Thus, although an evolved atomic note preserves its original timestamp, its revised fields are based solely on evidence available before prediction. This ensures that dynamic memory evolution does not inject future preference information into past memory states.

Table 9 summarizes the information available to each stage under this protocol.

Table 9: Leakage-control checklist for the temporal evaluation protocol.

Stage	Pre-t	Target	Post-t
Atomic Construction	✓	✗	✗
Semantic Linking	✓	✗	✗
Memory Evolution	✓	✗	✗
Evidence Synthesis	✓	✗	✗
Candidate Reranking	✓	Candidate	✗
Post-eval. Update	Delayed	–	Excluded

Notes. Pre-t and post-t denote interactions before and after the target interaction at time t. "Candidate" means that the target item is visible only as one candidate during reranking. "Delayed" means that the update happens after ranking, and "Excluded" means that it is not used for evaluating the same instance.

Table 10: Semantic link quality analysis on sampled links from the Books subset. Scores are rated on a 1–5 scale. Higher is better.

Method	Relatedness	Correctness	Usefulness
Embedding-only Links	3.72	3.41	3.28
AtomRec Links	**4.31**	**4.08**	**3.96**

Notes. Relatedness measures whether two linked notes are semantically related. Correctness measures whether the inferred relation is consistent with the linked notes. Usefulness measures whether the link provides useful evidence for recommendation.

H Semantic Link Quality Analysis

To further evaluate whether the semantic links constructed by AtomRec provide meaningful relational evidence, we conduct a link-level quality analysis on the Books subset. We compare the semantic links generated by AtomRec with an embedding-only linking baseline, where each new atomic note is connected to its top-k nearest notes according to cosine similarity without LLM-based semantic selection. This comparison allows us to examine whether the memory agent improves link quality beyond nearest-neighbor retrieval.

We sample 300 generated links from the Books analysis subset and evaluate each link along three dimensions: *Relatedness*, which measures whether the two linked notes are semantically related; *Relation Correctness*, which measures whether the inferred relation type or rationale is consistent with the linked notes; and *Recommendation Usefulness*, which measures whether the link provides useful evidence for downstream recommendation. Each dimension is rated on a 1–5 scale by an automatic judge using `gpt-4o`. We report the average score for each dimension in Table 10.

Table 11: Memory growth statistics of AtomRec. We report average atomic notes, semantic links, and evolved notes under the default setting.

Dataset	Notes/User	Notes/Item	Links/Note	Evolved/Inter.
Books	3.8	1.4	2.6	1.7
Goodreads	4.5	1.8	2.9	1.9
MovieTV	3.1	1.3	2.4	1.5
Yelp	2.9	1.2	2.1	1.3

Notes. Notes/User and Notes/Item denote the average number of atomic notes per user and item. Links/Note denotes the average number of semantic links per note. Evolved/Inter. denotes the average number of evolved notes per interaction.

Table 10 shows that AtomRec produces higher-quality links than the embedding-only baseline across all three dimensions. The improvement in relatedness indicates that LLM-guided link selection can filter out superficially similar but weakly useful neighbors. The gains in relation correctness and usefulness suggest that semantic linking provides more than similarity-based connectivity: it identifies why two memory notes are related and whether the relation can support downstream ranking. The improvement in recommendation usefulness further supports the role of semantic links as evidence for downstream ranking. These results complement the ablation study by showing that Semantic Collaborative Link Construction improves not only final recommendation accuracy, but also the intrinsic quality of the memory graph.

I Memory Growth Statistics

We further analyze the growth of the atomic collaborative memory graph. Since AtomRec stores fine-grained atomic notes and semantic links, it is important to examine whether the memory space grows excessively as interactions accumulate. Table 11 reports the average number of atomic notes per user, atomic notes per item, semantic links per note, and evolved notes per interaction across four datasets.

The memory size remains moderate across datasets. Books and Goodreads have more atomic notes per user because they contain longer and more content-driven user histories, while MovieTV and Yelp have fewer notes due to shorter or more context-dependent interaction patterns. The average number of links per note remains below three under the default linking setting, suggesting that the semantic memory graph does not become overly dense. The number of evolved notes per interaction is also limited, indicating that dynamic

Table 12: Detailed efficiency statistics per 1K interactions under the default lightweight backbone.

Method	Input Tok.	Output Tok.	Total Tok.	Cost / 1K
iAgent	5.8M	0.75M	6.5M	$1.35
MemRec	10.6M	1.50M	12.1M	$2.65
AtomRec	11.6M	1.80M	13.4M	$2.95

Table 13: Backbone-level relative cost index used in Figure 6. The default gpt-4o-mini backbone is normalized to 1.0×.

Backbone	Type	Relative Cost	Books N@5
Qwen3-8B	Open-source	0.35×	0.626
Qwen3-32B	Open-source	0.75×	0.663
gpt-4o-mini	Closed	1.00×	0.713
gpt-4o	Closed	16.7×	0.731
gpt-5.2	Closed	20.6×	0.736
gpt-5.4	Closed	23.0×	0.742

memory evolution updates a small set of related historical notes rather than repeatedly rewriting the entire memory space.

J Detailed Efficiency Statistics

This appendix provides the numerical statistics used in the efficiency analysis. We report token/-cost statistics under the default lightweight backbone and the relative cost index used in Figure 6. Token usage is measured from logged input and output tokens. Costs are estimated under the same pricing setting used in our experiments. For backbone comparison, we normalize each backbone cost by the default gpt-4o-mini setting, while Qwen3 costs are estimated from local serving rather than API pricing.

Table 12 shows that AtomRec uses more tokens than MemRec because it performs atomic construction, semantic linking, and field-level evolution. However, the increase remains moderate, and these memory-side operations can be cached and executed asynchronously. This supports the finding that AtomRec introduces **limited online reranking overhead**.

Table 13 shows that stronger closed backbones bring only modest additional Books N@5 gains after gpt-4o-mini, despite much higher relative costs. This supports the finding that the default lightweight backbone provides **a favorable cost-performance trade-off**. The cost values are intended to compare deployment regimes rather than provide universal pricing, since actual cost depends on batching, caching, hardware utilization, and provider-specific pricing.

K Prompt Templates

This appendix presents the prompt templates used by the memory and recommendation agents. Each prompt is implemented as a fixed instruction template with dynamic slots, such as `{user_id}` and `{candidate_notes}`, and corresponds to one stage of our atomic collaborative memory framework.

Stage-R: Collaborative Memory Retrieval Prompt. Stage-R retrieves collaborative memory evidence for the target user before recommendation. It takes the user's personal memory, collaborative neighbor memory cards, optional linked-memory synthesis, and candidate-item context as input, and outputs preference facets and support edges for downstream memory writing and reranking.

It outputs preference facets and support edges, which are later used by the memory writing module and the recommendation reranker.

Stage-R: Collaborative Memory Retrieval Prompt.

System Role. You are an intelligent collaborative memory retrieval system for personalized recommendation. Your task is to infer stable and emerging user preferences from atomized memories, including summary, context, keywords, tags, links, and collaborative neighbors.

Target User: User `{user_id}`

User's Personal Memory:
`{user_mem_bullets}`

Collaborative Neighbor Memory Cards:
The following neighboring users and items provide collaborative signals. Some entries include memory links to other notes; use them as evidence for broader context, while staying grounded in the provided neighbors.
`{neighbor_table_json}`

Context-Aware Collaborative Memory Synthesis:
`{collab_synth_text}`

Context Candidate Items:
`{candidates_json}`

Task. Analyze the user's personal memory and collaborative neighbor memory cards to identify `{n_facets}` preference facets. Prioritize facets supported by multiple signals, including user memory, collaborative neighbors, and linked memory evidence. Include both persistent preferences and emerging preferences.
For each preference facet, provide: (1) a concise natural-language description; (2) a confidence score between 0 and 1; and (3) supporting neighbors.
Also identify support edges between neighboring users/items and the target user, with edge weights between 0 and 1 indicating collaborative relevance strength.

Expected Output.
```
{"facets": [{"facet", "confidence",
"supporting_neighbors"}],
  "support_edges": [{"from", "to", "w"}]}
```

Stage-W: Collaborative Memory Writing Prompt. Stage-W updates memory after a new user–item interaction. It takes the current user memory, clicked item memory, extracted preference facets, and collaborative neighbors as input, and outputs updated user, item, and neighbor memories together with atomic fields, including keywords, tags, context, and link targets.

Stage-W: Collaborative Memory Writing Prompt

System Role. You are an intelligent memory management system for collaborative recommendation. Your task is to update the personal memories of the user, the clicked item, and relevant collaborative neighbors based on a new interaction.

Interaction Context.
User `{user_id}` has just interacted with Item `{item_id}`.
Clicked item information: `{clicked_item_info}`

User Preferences Extracted from Collaborative Memories.
`{preference_facets}`

Current User Memory.
`{current_user_memory}`

Current User Atomic Memory Card.
`{user_atomic_note}`

Current Item Memory.
`{current_item_memory}`

Current Item Atomic Memory Card.
`{item_atomic_note}`

Collaborative Neighbors Available for Memory Propagation.
`{neighbor_memory_cards}`

Task. Generate updated memories for: (1) the current user; (2) the clicked item; and (3) selected collaborative neighbors.
For user, item, and neighbor updates, also produce atomized fields: keywords, tags, context, and link targets. Use link targets to explicitly create collaborative semantic links. Prefer sparse and precise fields over long lists.

Expected Output.
```
{"user_memory": "...",
  "user_memory_atomic": {"keywords", "tags",
"context", "link_targets"},
  "item_memory": "...",
  "item_memory_atomic": {"keywords", "tags",
"context", "link_targets"},
  "neighbor_updates": [...]}
```

Semantic Collaborative Link Construction Prompt. This prompt supports Semantic Collaborative Link Construction. Given a newly created atomic memory note and top-k candidate notes retrieved by embedding similarity, it asks LM_Mem to select semantically related notes and produce link rationales, which are used to build the collaborative memory network.

Semantic Collaborative Link Construction Prompt.

System Role. You are LM_Mem for semantic collaborative link construction. Your task is to decide semantic links for a newly created atomic memory note by analyzing candidate notes retrieved by embedding similarity.

New Atomic Memory Note.
`{new_note}`

Candidate Memory Notes.
`{candidate_notes}`

Task. Select up to `{top_k_links}` candidate notes that should be semantically linked to the new note. Prefer links with one of the following relations: shared topic or theme; complementary preference; causal or intent progression; and cross-domain transfer.
Do not link notes that are only superficially similar.

Expected Output.
```
{"selected_note_ids": [...],
  "rationales": [...]}
```

Dynamic Memory Evolution Prompt. This prompt performs dynamic memory evolution when a new note is added. It provides LM_Mem with the new note, a target historical note, and nearby contextual notes, and asks whether the historical note should be updated to reflect newly emerging preference semantics.

Dynamic Memory Evolution Prompt

System Role. You are LM_Mem for dynamic memory evolution. Your task is to evolve an existing atomic memory note using a newly added note and nearby contextual notes.

New Note as Trigger.
`{new_note}`

Target Historical Note to Evolve.
`{target_note}`

Nearby Context Notes.
`{nearby_notes}`

Task. If evolution is beneficial, produce an updated memory text and updated atomic fields for the target note. Focus on keyword strengthening, tag generalization or refinement, and one-sentence context reconstruction reflecting preference evolution.
Keep the updated memory text concise and semantically consistent with the original target note. If no evolution is needed, return `should_update=false` and preserve the target semantics conservatively.

Expected Output.
```
{"should_update": true/false,
  "updated_memory": "...",
  "updated_atomic": {"keywords", "tags",
"context", "link_targets"},
  "reason": "..."}
```

Context-Aware Collaborative Synthesis Prompt. This prompt compresses a linked memory subgraph into collaborative evidence for recommendation. It takes the current user memory

and a two-hop linked memory subgraph as input, and outputs a compact summary, facet hints, and evidence paths for Stage-R and the downstream recommender.

Context-Aware Collaborative Synthesis Prompt

System Role. You are LM_Mem for context-aware collaborative retrieval synthesis. You are given a two-hop linked memory subgraph expanded from initially retrieved memories.

User Memory.
`{user_memory_text}`

Compressed Linked Memory Evidence.
`{compact_subgraph}`

Task. Synthesize collaborative evidence into a compact structured summary for the recommender LLM. Capture core themes, emerging extensions, and evidence-chain relations across linked memories.

Expected Output.
```
{"summary": "...",
  "facets_hint": [...],
  "evidence_paths": [...]}
```

Recommendation Reranking Prompt. This prompt is used by the final recommendation agent to score candidate items. It takes personal memory, collaborative evidence, preference facets, current user request, and candidate item memories as input, and outputs item-level relevance scores with brief rationales.

Recommendation Reranking Prompt

System Role. You are an intelligent recommendation scoring system. Your task is to rank candidate items for the target user based on personal memory and collaborative preference signals.

Target User. User `{user_id}`

User's Personal Memory.
`{user_memory_summary}`

Collaborative Evidence from Linked Memory Network.
`{collab_synth_text}`

User's Current Request.
`{instruction}`

User Preference Facets.
`{preference_facets}`

Candidate Item Memories.
`{candidate_item_memories}`

Task. Rank all candidate items by relevance to this user. Assign each item a score between 0 and 1. The best match should receive a high score, and no two items should share the same score. Primary preference facets should drive the top ranking more than secondary facets.
For each item, provide a brief rationale explaining the score based on the user's facets and memory.

Expected Output.
```
{"scores": [{"item_id", "score",
"rationale"}]}
```

L Additional Qualitative Analysis

We provide additional qualitative examples to complement the main case study. These examples include both successful and challenging scenarios, showing that atomic collaborative memory can support accurate recommendation when the evolved memory aligns with the target, but may still struggle with over-specific memory compression, semantically adjacent false positives, idiosyncratic short-term target deviations, and imperfect preservation of explicit collaborative links.

Case A: Over-Narrowed Memory. We first examine a challenging case where the model captures the correct broad preference region but compresses the user's memory into an overly specific subtheme. This case is useful because the retrieved and ranked items are not irrelevant; instead, the error comes from insufficient fine-grained discrimination within a semantically adjacent candidate set.

Case A: Over-Narrowed Memory Despite Correct High-Level Preference

User. User 6711

Final Memory Snapshot.

- **Note ID:** `N-190769`
- **Revision:** 3
- **History Reference:** `N-190765`
- **Links:** none

Memory Content. *User 6711 is deeply engaged in exploring themes of grief and healing, particularly in the context of child loss. They are actively seeking literature that provides support and understanding for navigating these profound experiences, including insights from works like "And a Sword Shall Pierce Your Heart" and "Gone but Not Lost". Their recent interest in "The Afterlife Interviews: Volume I" suggests a desire for diverse perspectives on grief and the afterlife.*

Atomic Fields.

- **Keywords:** grief, healing, child loss, support, literature, understanding, afterlife
- **Tags:** grief support, mental health, parenting, child loss, healing, exploration, afterlife
- **Context:** User is focused on grief and healing literature, especially related to child loss, seeking support and understanding for navigating profound experiences, now including afterlife perspectives.

Prediction. The ground-truth item is *On the Edge of the Etheric*, ranked at position 3. The two higher-ranked items are *BrokenHearted* and *Angels of Light Cards*.

Diagnosis. This is not a completely wrong recommendation. The evolved memory correctly captures a broad cluster around grief, healing, afterlife, and spiritual support. However, it becomes overly concentrated on the child-loss and grief-support subtheme. The user's broader history also contains stronger metaphysical and afterlife-oriented signals, such as *Proof of Heaven*, *The Afterlife Experiments*, *Destiny of Souls*, *The Last Frontier*, and *The Afterlife Interviews*. As a result, the reranker favors generic emotional or spiritual support books over the more directly etheric and afterlife-oriented target.

Takeaway. Memory evolution helps identify the correct high-level semantic region, but the final memory can become too narrow around one dominant subtheme. This leads to insufficient fine-grained discrimination between generic spiritual-healing candidates and directly metaphysical or etheric targets.

Case B: Idiosyncratic Target Deviation. The second case illustrates a different failure mode: the evolved memory is coherent and well supported by the user's dominant history, but the ground-truth item reflects a short-term or idiosyncratic deviation. This highlights the tension between stable long-term preference modeling and exploratory user behavior.

Case B: Idiosyncratic Target Deviation from Dominant Preference

User. User 5356

Final Memory Snapshot.

- **Note ID:** `N-190770`
- **Revision:** 2
- **History Reference:** `N-190758`
- **Links:** `N-190775`, `N-174450`, `N-190771`, `N-46822`
- **Evolution Log:** from `N-190775`, score 0.785

Memory Content. *A user interested in personal development and psychological insights, particularly through the lens of Buddhist teachings. Enjoys literature that offers wisdom and understanding of the human experience, and is now exploring visual storytelling through unique editions like stereoscopic books.*

Atomic Fields.

- **Keywords:** personal development, Buddhist teachings, psychology, wisdom, human experience, visual storytelling, stereoscopic books
- **Tags:** self-help, spirituality, Buddhist, personal growth, visual arts
- **Context:** User seeks literature that provides psychological insights and personal growth, now showing interest in visual storytelling.

Prediction. The ground-truth item is *The King of Style: Dressing Michael Jackson*, ranked at position 3. The two higher-ranked items are *Teresa of Avila: The Progress of a Soul* and *Victory in Singleness: A Strategy for Emotional Peace*.

Diagnosis. The model does not make an unreasonable prediction. The final memory strongly emphasizes Buddhist teachings, psychology, personal development, spirituality, and personal growth. It also captures a weaker signal about visual storytelling and stereoscopic books, which suggests that the memory is trying to absorb later visual-format interests. However, the target item about Michael Jackson's style is a sharper shift toward fashion, music culture, and visual celebrity aesthetics. The model therefore follows the dominant long-term preference trajectory rather than this idiosyncratic short-term deviation.

Takeaway. This case exposes a tension between long-term memory consistency and short-term exploration. The evolved memory is stable and coherent, but the ground-truth item reflects a transient preference shift that is weakly supported by the user's prior history.

Case C: Successful Memory–Target Alignment. We also include a successful example as a contrast to the previous two cases. Here, the evolved memory center aligns closely with the target item, showing that atomic abstraction and memory evolution can support accurate top-ranked recommendation when the target follows the user's dominant preference trajectory.

Case C: Successful Alignment Between Evolved Memory and Target

User. User 4991

Final Memory Snapshot.

- **Note ID:** `N-190773`
- **Revision:** 3
- **History Reference:** `N-190771`
- **Links:** `N-167292`, `N-142015`, `N-177460`, `N-149729`

Memory Content. *User 4991 is interested in personal stories of resilience and transformation, particularly those involving near-death experiences and life reflections. They appreciate narratives that explore profound life changes and the human experience, and are now also drawn to themes of manifestation, personal empowerment, and psychological insights through the lens of Buddhist teachings, including visual storytelling.*

Atomic Fields.

- **Keywords:** personal stories, resilience, transformation, near-death experience, life reflections, manifestation, empowerment, psychology, Buddhist teachings, visual storytelling
- **Tags:** biography, self-discovery, personal growth, transformation, spirituality, visual arts

Prediction. The ground-truth item is *Driving Straight on Crooked Lines*, ranked at position 1.

Diagnosis. This case shows when memory evolution works well. The target item aligns with the user's evolved memory center, including personal narrative, resilience, transformation, self-discovery, and spiritual reflection. Unlike User 6711, the memory abstraction is not compressed into an overly narrow subtheme. Unlike User 5356, the target does not deviate sharply from the dominant preference trajectory.

Takeaway. When the target item is well aligned with the evolved memory center, atomic memory abstraction and collaborative evolution can support accurate top-ranked recommendation.

Additional Observation: Semantic Absorption vs. Link Traceability

Observed Link Statistics.

- **User 6711:** `cross_user_links = 0, cross_item_links = 0`
- **User 5356:** `cross_user_links = 1, cross_item_links = 3`
- **User 4991:** `cross_user_links = 0, cross_item_links = 4`

Observation. For User 6711, the latest memory does not preserve explicit collaborative links, even though earlier traces suggest that collaborative evidence contributes to memory evolution.

Diagnosis. This suggests that collaborative signals may be absorbed into memory text, keywords, tags, and contextual descriptions, but are not always retained as explicit final-step links. In other words, collaborative evidence can shape memory semantics while becoming less traceable in the final memory graph.

Implication. Future work could improve structural fidelity by introducing link persistence, confidence-aware link decay, and provenance tracking during memory evolution.

These cases reveal three complementary behaviors. User 4991 shows a successful case where the evolved memory center aligns with the target item, yielding a top-1 hit. User 6711 shows that memory evolution can capture the correct broad preference region but over-compress it into a narrow subtheme, causing semantically adjacent false positives. User 5356 shows that stable long-term memory may under-rank idiosyncratic target items that reflect short-term deviations. Together, these cases suggest that future work should improve fine-grained reranking, preserve explicit collaborative provenance, and better distinguish persistent preference evolution from exploratory behavior.

M Additional Limitations and Deployment Considerations

Privacy and Governance. ATOMREC builds semantic links across user and item memories, which may raise privacy and governance concerns in real deployments. Although our experiments use public benchmark data and do not involve personally identifiable information, fine-grained atomic notes may still encode sensitive preference traces. Practical systems should support data minimization, user-level deletion, provenance tracking, and access control for cross-user links. One possible deployment option is to restrict semantic linking within privacy-preserving user groups or user-siloed settings, where only aggregated or anonymized memory evidence can be shared across users.

Latency and Deployment. Our efficiency analysis focuses on token cost and average memory statistics rather than end-to-end serving latency under concurrent traffic. Real latency depends on API infrastructure, batching, caching, deployment hardware, and whether memory-side operations are executed online or asynchronously. In practice, atomic construction, semantic linking, and memory evolution can be performed after interactions and cached for later recommendation. Future work should evaluate p50/p90 latency and throughput in online serving environments.

Backbone Dependence. We primarily evaluate ATOMREC with `gpt-4o-mini`. Although the framework is model-agnostic in design, different open-source backbones may produce different link, evolution, and reranking quality. This dependence may affect reproducibility for groups without access to the same proprietary model. Evaluating stronger open-source memory and recommendation agents is an important direction for future work.

N Extended Related Work

This section provides a comprehensive review of the literature pertinent to our framework, detailing the evolution of memory mechanisms in Large Language Models (LLMs) and the trajectory of LLM-based autonomous agents in recommender systems.

Memory Mechanisms in LLM Agents. While LLMs excel in long-horizon reasoning (Achiam et al., 2023; Ouyang et al., 2022; Brown et al., 2020; Yao et al., 2022), their restricted context windows necessitate external memory for persistent knowl-

edge retention (Hu et al., 2025; Zhang et al., 2025b; Du, 2026; Park et al., 2023; Shinn et al., 2023). Early architectures rely on static workflows, such as MemoryBank's forgetting curves (Zhong et al., 2024), MemGPT's hierarchical buffers (Packer et al., 2023), and SCM's read-write streams (Wang et al., 2023). To enhance adaptability, recent systems (e.g., A-Mem (Xu et al., 2026), Mem0 (Chhikara et al., 2025), MemInsight (Salama et al., 2025)) transition to dynamic paradigms via associative linking and retrospective summarization (Fang et al., 2025; Li et al., 2025). However, these designs overwhelmingly optimize for isolated, single-agent environments, leaving collaborative memory—where multiple agents co-evolve an interconnected memory space—largely uncharted for interactive recommendation.

Large Language Models for Recommendation (LLM4Rec). Before the advent of autonomous agents, research adapted LLMs for recommendation through various paradigms. Early approaches leveraged prompting (Chat-REC (Gao et al., 2023), zero-shot rankers (Hou et al., 2024)) or instruction tuning and task unification (P5 (Geng et al., 2022), InstructRec (Zhang et al., 2026b), TALLRec (Bao et al., 2023)) to align LLMs with domain-specific patterns. To resolve vocabulary mismatches, generative models introduced Semantic ID (SID) paradigms (TIGER (Rajput et al., 2023; Ju et al., 2025), LC-Rec (Zheng et al., 2024), MinioneRec (Kong et al., 2025)) to natively generate structured identifiers and enhance cross-domain generalization. Concurrently, efforts like Cot4Rec (Yue et al., 2025) and R2Rec (You et al., 2026) injected Chain-of-Thought (CoT) reasoning. Despite these advancements, traditional LLM4Rec models remain static, single-turn predictors lacking the autonomy, interactive tool-use, and evolving memory required for dynamic environments, thus catalyzing the shift towards agentic systems.

Agents-based Recommendation. The integration of agentic capabilities into recommender systems has shifted the paradigm from passive modeling to active planning (Peng et al., 2025; Zhang et al., 2025a; Zhu et al., 2025), predominantly categorized into simulation and recommender-oriented approaches. Simulation frameworks deploy agents as digital twins to model interaction dynamics; for instance, Agent4Rec (Zhang et al., 2024a) uses agents as user simulators, while AgentCF (Zhang et al., 2024b) proposes a bidirectional multi-type (user-item) agent simulation. Conversely, recommender-oriented systems design agents to execute complex tasks. While initial efforts focused on empowering a single centralized agent through self-inspired planning (RecMind (Wang et al., 2024a)) or brain-toolbox architectures (InteRecAgent (Huang et al., 2025)), recent pioneering works like MACRec (Wang et al., 2024b) enable adaptable multi-agent collaboration to handle diverse and complex user intents.

Bridging the Gap: Memory in Agentic RS. To sustain long-term personalization, integrating explicit memory into agentic RS is essential, yet most systems update memories in isolation. For example, iAgent (Xu et al., 2025) and RecBot (Tsitseklis et al., 2023) confine updates to individual user profiles via self-reflection, entirely discarding high-order collaborative connectivity. To address this, the state-of-the-art framework MemRec (Chen et al., 2026) constructs a macroscopic collaborative memory graph. However, MemRec fundamentally suffers from coarse-grained, node-level representations that obfuscate specific intents, and rigid graph propagation rules bounded by explicit historical interactions. In contrast, our proposed framework comprehensively deconstructs monolithic memories into multi-attribute *atomic notes* and replaces predefined edges with LLM-driven *autonomous semantic linkage*, achieving dynamic memory evolution at a granular level and providing a structured basis for context-aware collaborative reasoning.

Generative AI Statement

Generative AI tools were used only for language polishing and grammar checking. All technical contributions, experiments, analyses, and conclusions were developed and verified by the authors, who take full responsibility for the paper.