ALLOCATING AGENCY: STRUCTURE-PRESERVING MULTI-AGENT REASONING FOR RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have introduced a new paradigm of **agentic recommendation**, yet effectively integrating LLM agents into mature recommender systems remains challenging. Existing recommendation models already encode substantial intelligence through collaborative signals, retrieval mechanisms, and behavioral priors, making unrestricted LLM autonomy an insufficient solution. We formulate this challenge as the **Agency Allocation Problem**: how to assign LLM reasoning authority to unresolved recommendation decisions while preserving the intelligence already captured by specialized recommenders. Based on this perspective, we propose **Agent-R**, a structure-preserving framework that allocates LLM agency across two complementary decision spaces. In the **Consideration Space**, a Planning Agent coordinates heterogeneous recommendation evidence to adapt candidate coverage around a reliable retrieval backbone. In the **Preference Space**, a Ranking Agent learns contextual preference residuals over an explicit recommendation prior to refine ranking decisions without replacing existing structures. Experiments across diverse recommendation domains demonstrate that Agent-R consistently improves both candidate coverage and ranking quality, achieving up to 28.3% relative gains in Recall@10 over strong fusion baselines. Further analyses show that structured agency allocation enables LLMs to provide complementary reasoning while retaining the strengths of specialized recommendation models. Code is available at `https://anonymous.4open.science/r/agentr_repo-E5E5/`.

1 INTRODUCTION

Recent advances in large language models (LLMs) have motivated a new paradigm of **agentic recommendation**, where LLM-based agents extend recommender systems beyond static preference prediction by reasoning over users, items, and heterogeneous recommendation evidence Wang et al. (2023); Zhang et al. (2024a); Huang et al. (2025d). By leveraging contextual understanding, semantic knowledge, and interaction history, LLM agents provide a promising direction for building more adaptive and personalized recommendation systems Geng et al. (2022); Bao et al. (2023); Zhu et al. (2024). However, introducing LLM agents into recommendation also creates a fundamental challenge: modern recommender systems are already highly structured decision-making systems that encode substantial intelligence through collaborative signals, retrieval mechanisms, behavioral patterns, and domain-specific inductive biases Covington et al. (2016); Liu et al. (2022). Therefore, the key challenge is not simply *whether LLMs can make recommendation decisions*, but *how LLM reasoning should interact with and complement the intelligence already embedded in existing recommendation pipelines*.

A common direction in agentic recommendation is to increase LLM autonomy by allowing LLMs to directly generate candidates, rank items, or coordinate recommendation decisions Wang et al. (2023); Zhang et al. (2024a); Huang et al. (2025d). Although this paradigm benefits from the broad reasoning capability of LLMs, it overlooks an important property of mature recommender systems: specialized models already capture valuable recommendation structures that are difficult to reconstruct from general-purpose reasoning alone Geng et al. (2022); Zhu et al. (2024). For example, collaborative models encode behavioral regularities from historical interactions, retrieval systems provide efficient access to semantically relevant candidates, and ranking models preserve statistical preferences learned from large-scale user–item interactions Covington et al. (2016); Liu et al.

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

(2022). Consequently, increasing LLM autonomy alone is insufficient and may even discard reliable recommendation intelligence already present in the system. This motivates a fundamental research question for agentic recommendation: the **Agency Allocation Problem**, which asks *how LLM decision authority should be assigned to unresolved recommendation decisions while preserving the intelligence encoded by specialized models*.

Within recommendation pipelines, this problem naturally manifests through multiple decision stages with different forms of existing intelligence and unresolved decisions Huang et al. (2025b;c). Modern recommender systems simultaneously determine which items should enter the decision process and how those items should be prioritized, where each stage requires different forms of reasoning and intervention Covington et al. (2016); Huang et al. (2025b). The first decision space is the **Consideration Space**, which determines the set of items exposed to downstream ranking. A failure in this space occurs when potentially relevant items are absent from the candidate set and therefore cannot be recovered by subsequent ranking. The second decision space is the **Preference Space**, which determines how considered items should be prioritized. A failure in this space occurs when relevant items are already available but remain insufficiently exposed due to missing contextual preference information Liu et al. (2022). These two spaces therefore define distinct **unresolved decision spaces**: the former concerns *what can be considered*, while the latter concerns *how considered items should be prioritized*. Effective agentic recommendation consequently requires allocating LLM agency according to the nature of each decision space rather than granting uniform autonomy throughout the recommendation pipeline.

Motivated by this perspective, we propose **Agent-R**, a multi-agent recommendation framework that instantiates agency allocation through two complementary decision processes. For the **Consideration Space**, Agent-R enables an LLM-based planner to coordinate heterogeneous recommendation evidence around a reliable candidate backbone, where the LLM determines how complementary signals should be incorporated rather than directly selecting items. For the **Preference Space**, Agent-R leverages structured user memory and recommendation context to learn a contextual preference residual over an explicit recommendation prior, allowing LLM reasoning to refine rather than replace existing ranking structures. These two mechanisms address complementary limitations of existing recommendation systems: improving what can be considered through evidence coordination and improving how considered items are prioritized through contextual preference refinement. By assigning LLM reasoning to different stages of the recommendation process, Agent-R combines the contextual understanding of LLMs with the structural strengths of specialized recommenders.

Our main contributions can be summarized as follows:

- We formulate agentic recommendation as an **Agency Allocation Problem**, showing that effective LLM integration requires assigning reasoning authority to unresolved decision spaces while preserving the intelligence already encoded by specialized recommenders.
- We propose **Agent-R**, a structure-preserving framework that allocates LLM agency in two complementary spaces: evidence coordination for consideration-space adaptation and contextual residual refinement for preference-space personalization.
- Comprehensive experiments across diverse recommendation domains demonstrate consistent improvements in candidate coverage and ranking quality, while controlled analyses validate the effectiveness of structured LLM intervention over unrestricted recommendation autonomy.

2 Related Work

Recent advances in large language models (LLMs) have motivated extensive studies on LLM-enhanced and agentic recommendation systems. Early works explore LLMs as recommendation models through language generation and instruction tuning, such as P5 Geng et al. (2022), OpenP5 Xu et al. (2024), TALLRec Bao et al. (2023), and LLM4Rec Zhu et al. (2024), while later approaches leverage LLMs to enhance conventional recommenders through semantic augmentation and representation alignment Wei et al. (2024); Ren et al. (2024). More recent agentic recommendation frameworks, including RecAgent Wang et al. (2023), InteRecAgent Huang et al. (2025d), Agent4Rec Zhang et al. (2024a), AgentCF Zhang et al. (2024b), and MACRec Wang et al. (2024b), introduce planning, memory, and multi-agent collaboration into recommendation pipelines. How-

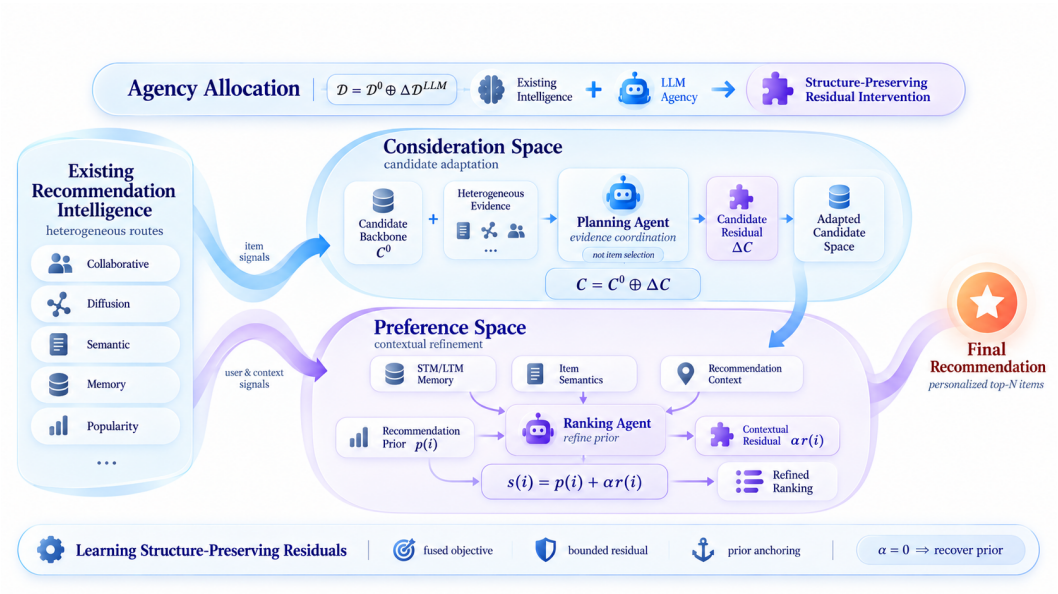

Figure 1: Overview of **Agent-R**. The framework preserves reliable structures from specialized recommenders while allocating LLM reasoning to two unresolved decision spaces through structure-preserving agency allocation. A Planning Agent performs bounded consideration-space agency allocation by coordinating heterogeneous recommendation evidence, and a Ranking Agent refines the inherited recommendation prior through contextual preference residuals.

ever, existing approaches primarily focus on improving LLM capability or increasing agent autonomy, while overlooking how LLM decision authority should be allocated when recommendation systems already contain specialized intelligence.

A complementary line of work studies how to combine heterogeneous recommendation routes and ranking experts. Traditional fusion methods such as CombMNZ Fox & Shaw (1994), Reciprocal Rank Fusion (RRF) Cormack et al. (2009), and learning-to-rank based fusion methods such as LambdaMART Burges (2010); Wu et al. (2010) aggregate multiple recommendation signals through fixed rules or learned combination functions. Recent multi-channel retrieval and fusion approaches further explore adaptive route weighting and reranking strategies Covington et al. (2016); Huang et al. (2025b); Zhang et al. (2025); Huang et al. (2025c). Nevertheless, these methods treat recommendation routes as passive information sources and focus on learning better aggregation functions. In contrast, Agent-R studies *where LLM agents should receive decision authority* and introduces a structure-preserving agency allocation framework that coordinates heterogeneous evidence and refines unresolved recommendation decisions.

3 METHOD

In this section, we introduce **Agent-R**, a structure-preserving multi-agent LLM framework that instantiates the agency allocation formulation for agentic recommendation. An overview of Agent-R is shown in Fig. 1, where different LLM agents are assigned to different recommendation decision spaces. The following sections describe how Agent-R allocates LLM agency in the consideration and preference spaces.

3.1 AGENCY ALLOCATION FRAMEWORK

Agent-R formulates agentic recommendation as an **agency allocation** problem, where LLM agents refine rather than replace the intelligence encoded by specialized recommendation models. Let $\mathcal{D}_u^0$ denote the decision state produced by existing recommenders, capturing reliable recommendation structures such as candidate exposure, ranking priors, and heterogeneous evidence. Instead of gen-

erating a new decision from an LLM, Agent-R introduces LLM reasoning as a structure-preserving residual intervention:

$$\mathcal{D}_u = \mathcal{D}_u^0 \oplus \Delta\mathcal{D}_u^{\mathrm{LLM}}, \tag{1}$$

where $\Delta\mathcal{D}_u^{\mathrm{LLM}}$ denotes the decision residual assigned to LLM agents and $\oplus$ represents a structure-preserving composition operator. This formulation allows LLM agents to provide complementary reasoning only where existing recommendation intelligence remains insufficient.

In recommendation, such residual interventions arise from two heterogeneous decision spaces:

$$\Delta\mathcal{D}_u^{\mathrm{LLM}} = \{\Delta\mathcal{C}_u, \Delta s_u\}, \tag{2}$$

where $\Delta\mathcal{C}_u$ corresponds to consideration-space adaptation and Δs_u corresponds to preference-space refinement. Accordingly, Agent-R allocates LLM agency through two complementary mechanisms:

$$\underbrace{\mathcal{C}_u^0 \to \mathcal{C}_u^0 \oplus \Delta\mathcal{C}_u}_{\textbf{Consideration-Space Agency Allocation}} \quad \text{and} \quad \underbrace{p_u(i) \to p_u(i) + \alpha r_\theta(u,i)}_{\textbf{Preference-Space Agency Allocation}}. \tag{3}$$

The former employs LLM-guided evidence coordination for candidate adaptation, while the latter uses contextual residual modeling for preference refinement over an explicit recommendation prior. The following sections detail these two forms of LLM agency allocation.

3.2 Allocating Agency in Consideration Space

The consideration space determines which items enter the downstream recommendation process. Existing recommender systems already provide a reliable candidate backbone through collaborative filtering, graph-based models, semantic retrieval, and other specialized retrieval mechanisms. However, different routes capture different inductive biases and may expose complementary candidate regions. Therefore, the unresolved decision in this space is not whether an LLM can directly select items, but *how heterogeneous recommendation evidence should be coordinated around an existing candidate backbone*.

Let $\mathcal{C}_u^0$ denote the candidate set generated by the base recommender for user u. Agent-R allocates consideration-space agency through a **Planning Agent**, which introduces a bounded candidate residual:

$$\mathcal{C}_u = \mathcal{C}_u^0 \oplus \Delta\mathcal{C}_u, \tag{4}$$

where $\Delta\mathcal{C}_u$ denotes complementary candidates introduced through LLM-guided evidence coordination. The Planning Agent does not perform item-level candidate selection; instead, it determines how recommendation evidence should be coordinated around the existing candidate backbone.

To enable evidence-level coordination, Agent-R constructs a structured recommendation state

$$\mathcal{S}_u^{\mathrm{cand}} = \left(m_u^{\mathrm{str}}, \{\widetilde{\mathbf{e}}_{u,i}\}_{i\in\mathcal{U}_u}, \bar{\mathbf{w}}\right), \tag{5}$$

where m_u^{str} summarizes structural user preferences, $\widetilde{\mathbf{e}}_{u,i}$ represents calibrated evidence from heterogeneous recommendation routes, and $\bar{\mathbf{w}}$ provides a global reliability reference across evidence channels. This state exposes recommendation intelligence to the Planning Agent while preserving the provenance of different evidence sources.

Given this state, the Planning Agent operates over a bounded policy space rather than individual candidate decisions:

$$\pi_u^{\mathrm{con}} = A_{\mathrm{plan}}(\mathcal{S}_u^{\mathrm{cand}}) = (b_u, \mathcal{A}_u, k_u, d_u, B_u, \boldsymbol{\delta}_u), \tag{6}$$

where the policy specifies the base route, complementary routes, protected candidate depth, intervention budget, and evidence adjustments. The adjustment variables are restricted to a compact action space, $\delta_{u,j} \in \{0.5, 1, 2\}$, corresponding to down-weighting, preserving, or emphasizing different recommendation signals. Thus, the Planning Agent controls evidence coordination, while specialized recommendation components remain responsible for item-level execution.

Given π_u^{con}, the executor preserves a reliable prefix of the base route and introduces complementary candidates supported by the selected evidence sources. After filtering candidates with insufficient cross-route support, it computes the calibrated evidence score $\eta_u(i) = \sum_j w_{u,j}\widetilde{e}_{u,i}^{(j)}$, where the planner-conditioned weights adjust the reference reliability $\bar{\mathbf{w}}$ according to $\boldsymbol{\delta}_u$. The residual candidate set is selected as

$$\Delta\mathcal{C}_u = \mathrm{Top}_{B_u}\{\eta_u(i)\}, \tag{7}$$

and composed with the original candidate backbone through Eq. equation 4.

3.3 Allocating Agency in Preference Space

While consideration-space agency determines which items enter the recommendation process, preference-space agency determines how these items should be prioritized. Existing recommenders already capture strong ranking structures through collaborative signals, behavioral patterns, and learned user–item interactions. However, such ranking priors may leave contextual preference factors unresolved, including evolving interests and semantic compatibility. Therefore, preference-space agency allocation focuses on *refining existing ranking intelligence* rather than replacing it with an independent LLM-generated ranking.

Given the adapted candidate space $\mathcal{C}_u$, Agent-R employs a **Ranking Agent** to refine the inherited ranking structure. The Ranking Agent first converts the pre-refinement ordering into an explicit recommendation prior:

$$p_u(i) = \phi\left(\mathrm{rank}_u^0(i)\right), \qquad i \in \mathcal{C}_u,$$

where $\phi(\cdot)$ preserves the original ranking order. Instead of reconstructing the complete ranking function, the Ranking Agent models only the unresolved preference component through a contextual residual:

$$s_u(i) = p_u(i) + \alpha r_\theta(u, i), \tag{8}$$

where $r_\theta(u, i)$ captures additional user–item compatibility beyond the inherited prior and α controls the intervention strength. Thus, the LLM refines the ranking structure rather than replacing the recommendation intelligence already encoded by specialized models.

To estimate the residual, the Ranking Agent considers three complementary sources of information: user preference memory, item semantics, and recommendation context. For each user–candidate pair, the input representation is

$$z_{u,i} = \mathrm{Serialize}(M_u, x_i, \rho_{u,i}), \tag{9}$$

where M_u, x_i, and $\rho_{u,i}$ denote structured user preferences, item information, and recommendation evidence, respectively. The residual is obtained through a bounded LLM-based projection:

$$r_\theta(u, i) = \tanh\left(\mathbf{v}^\top \mathrm{LN}(\mathrm{LLM}_\theta(z_{u,i})) + b\right) \in [-1, 1]. \tag{10}$$

The bounded residual constrains LLM intervention and ensures that contextual reasoning complements rather than overwhelms the inherited recommendation prior.

The final ranking can therefore be interpreted as a decomposition between existing recommendation intelligence and LLM residual agency:

$$\boxed{s_u(i) = \underbrace{p_u(i)}_{\text{existing recommendation intelligence}} + \underbrace{\alpha r_\theta(u, i)}_{\text{LLM residual agency}}}. \tag{11}$$

This formulation assigns the Ranking Agent a specific role in the preference space: providing contextual preference information that complements, rather than replaces, the ranking structures learned by specialized recommendation models.

3.4 Learning Structure-Preserving Residuals

The two agency allocation mechanisms introduce LLM interventions into different recommendation decisions. To ensure that these interventions complement rather than replace existing recommendation intelligence, Agent-R learns residuals through the fused recommendation objective while explicitly preserving the inherited recommendation structure.

For a user u with observed relevant item y_u, Agent-R optimizes the fused recommendation score:

$$\mathcal{L}_{\mathrm{rank}} = -\log \frac{\exp(s_u(y_u)/\tau)}{\sum_{i \in \mathcal{C}_u} \exp(s_u(i)/\tau)}, \tag{12}$$

where

$$s_u(i) = p_u(i) + \alpha_{\mathrm{train}} r_\theta(u, i).$$

Optimizing the fused decision rather than the residual alone encourages the LLM to learn contextual preference information relative to the existing recommendation prior, instead of reconstructing the ranking function from scratch.

The residual intervention is bounded as $r_\theta(u, i) \in [-1, 1]$, with its influence controlled by the fusion coefficient α. This provides an anchoring property:

$$\alpha = 0 \Longrightarrow \mathrm{rank}_{s_u} = \mathrm{rank}_{p_u}, \tag{13}$$

meaning that removing LLM intervention exactly recovers the original recommendation structure. Therefore, LLM reasoning acts as a controlled refinement over existing intelligence rather than an unconstrained replacement.

To stabilize residual learning, Agent-R regularizes excessive residual activation and optimizes the final objective:

$$\mathcal{L} = \sum_{u \in \mathcal{D}} \left(\mathcal{L}_{\mathrm{rank}} + \beta \mathcal{L}_{\mathrm{reg}} \right), \tag{14}$$

where $\mathcal{L}_{\mathrm{reg}}$ penalizes overly large residual responses. At inference time, the residual strength α is selected on validation data to control the degree of LLM intervention.

4 Experiments

Experimental Setup. We evaluate Agent-R on five recommendation benchmarks covering diverse domains, including Software, Industrial, DoubanBook, KuaiRec, and MicroLens. We compare with specialized recommenders, multi-route fusion methods, and LLM-based ranking baselines under the same chronological evaluation protocol. We report Recall@10/50 and NDCG@10/50, while complete dataset statistics, implementation details are provided in Appendix B.1.

4.1 RQ1: Overall Performance

Table 1 compares Agent-R with three categories of baselines: specialized recommendation models, heterogeneous route-based methods, and LLM-based fusion approaches. Compared with specialized recommenders, Agent-R consistently improves recommendation quality across domains, demonstrating that LLM agents can provide complementary value on top of existing recommendation intelligence. More importantly, Agent-R outperforms conventional fusion strategies that receive the same heterogeneous recommendation routes, indicating that the gains come from structured agency allocation rather than simply aggregating more evidence. For example, on KuaiRec, Agent-R improves R@10/N@10 from 0.0750/0.0500 achieved by the strongest fusion baseline to 0.0962/0.0604. On MicroLens, it improves the corresponding metrics from 0.0290/0.0143 to 0.0371/0.0176. The improvement is also consistent on domains with stronger recommendation priors, such as Software, where Agent-R improves the strongest fusion baseline from 0.1400/0.0782 to 0.1506/0.0845 on R@10/N@10. Finally, direct LLM fusion underperforms Agent-R despite accessing the same recommendation evidence, demonstrating that introducing LLMs alone is insufficient. Effective agentic recommendation requires assigning LLM agency to complementary decision spaces while preserving reliable recommendation structures.

4.2 RQ2: Ablation Study

Table 2 validates the functional separation of different agency allocation mechanisms in Agent-R. Removing candidate-space adaptation mainly degrades deeper candidate coverage, reducing R@50 from 0.3095 to 0.2980 on Software and from 0.0852 to 0.0800 on Industrial, while leaving head-ranking metrics largely unchanged. This behavior confirms that consideration-space agency primarily resolves candidate exposure failures by expanding what can be considered, rather than modifying the ranking preference over an existing candidate set. In contrast, removing the Ranking Agent keeps R@50 unchanged but substantially reduces head-ranking quality, with N@10 dropping from 0.0845 to 0.0762 and R@10 dropping from 0.1506 to 0.1341 on Software. The complementary effects of these two ablations support our decomposition of recommendation into distinct decision spaces, where candidate coverage and preference refinement require different forms of LLM agency. Furthermore, removing the recommendation prior causes a larger degradation than removing memory

Table 1: Overall recommendation performance across specialized recommendation models, single recommendation routes, and multi-route fusion methods. **Single Routes** are the heterogeneous recommendation routes used by Agent-R, while **Multi-Route Fusion** methods receive the same five route outputs as Agent-R under the identical evaluation protocol. The best result is shown in **bold**, and the second-best is underlined.

Dataset	Metric	Specialized Rec			Single Routes					Multi-Route Fusion					Ours
		BPR	LightGCN	SGL	NCL	DiffRec	Qwen-Emb	Mem-Ret	Pop	RRF	CombMNZ	Val-W	LambdaMART	LLM	**Agent-R**
Software	R@10	0.1154	0.1199	0.1200	0.1230	0.1341	0.0139	0.0287	0.0608	0.1334	0.1239	0.1341	0.1400	0.1253	**0.1506**
	N@10	0.0642	0.0678	0.0685	0.0700	0.0762	0.0070	0.0152	0.0327	0.0721	0.0650	0.0762	0.0782	0.0653	**0.0845**
	R@50	0.2748	0.2809	0.2822	0.2865	0.2980	0.0439	0.0732	0.1818	0.3040	0.3001	0.2980	0.3040	0.3040	**0.3095**
	N@50	0.1006	0.1048	0.1057	0.1075	0.1142	0.0138	0.0253	0.0608	0.1115	0.1056	0.1142	0.1160	0.1068	**0.1215**
Industrial	R@10	0.0219	0.0339	0.0316	0.0330	0.0322	0.0142	0.0140	0.0212	0.0398	0.0386	0.0368	0.0313	0.0350	**0.0418**
	N@10	0.0121	0.0186	0.0179	0.0187	0.0177	0.0079	0.0076	0.0112	0.0215	0.0212	0.0204	0.0189	0.0189	**0.0224**
	R@50	0.0603	0.0814	0.0797	0.0800	0.0726	0.0365	0.0395	0.0509	0.0909	0.0914	0.0905	0.0748	0.0909	**0.1045**
	N@50	0.0208	0.0293	0.0288	0.0293	0.0267	0.0127	0.0131	0.0178	0.0340	0.0331	0.0323	0.0287	0.0317	**0.0366**
KuaiRec	R@10	0.0400	0.0314	0.0146	0.0266	0.0329	0.0039	0.0017	0.0141	0.0244	0.0252	0.0329	0.0750	0.0312	**0.0962**
	N@10	0.0146	0.0120	0.0065	0.0110	0.0147	0.0016	0.0007	0.0052	0.0090	0.0089	0.0147	0.0500	0.0117	**0.0604**
	R@50	0.1254	0.1248	0.1010	0.1185	0.1203	0.0178	0.0093	0.1032	0.1277	0.1258	0.1203	0.1365	0.1277	**0.1391**
	N@50	0.0334	0.0326	0.0251	0.0311	0.0333	0.0046	0.0023	0.0240	0.0322	0.0316	0.0333	0.0632	0.0331	**0.0677**
MicroLens	R@10	0.0181	0.0233	0.0227	0.0215	0.0230	0.0230	0.0190	0.0024	0.0290	0.0289	0.0290	0.0286	0.0278	**0.0371**
	N@10	0.0084	0.0110	0.0105	0.0103	0.0106	0.0111	0.0095	0.0012	0.0143	0.0143	0.0142	0.0142	0.0132	**0.0176**
	R@50	0.0634	0.0784	0.0790	0.0726	0.0704	0.0569	0.0442	0.0088	0.0850	0.0866	0.0879	0.0766	0.0849	**0.0895**
	N@50	0.0182	0.0229	0.0226	0.0213	0.0210	0.0186	0.0150	0.0025	0.0265	0.0268	0.0272	0.0247	0.0257	**0.0276**
DoubanBook	R@10	0.0129	0.0169	0.0000	0.0169	0.0135	0.0158	0.0023	0.0174	0.0208	0.0191	0.0214	0.0169	0.0195	**0.0231**
	N@10	0.0074	0.0086	0.0000	0.0086	0.0066	0.0081	0.0014	0.0096	0.0100	0.0098	**0.0130**	0.0090	0.0112	0.0115
	R@50	0.0551	0.0574	0.0073	0.0585	0.0518	0.0360	0.0113	0.0512	0.0636	0.0619	0.0625	0.0478	0.0636	**0.0653**
	N@50	0.0163	0.0173	0.0015	0.0176	0.0149	0.0125	0.0032	0.0170	0.0192	0.0191	0.0208	0.0157	0.0201	**0.0215**

Table 2: Ablation study of Agent-R on Software and Industrial. Removing each component isolates the contribution of preference memory representation, consideration-space agency allocation, preference-space residual refinement, and recommendation prior anchoring.

Variant	Software				Industrial			
	N@10	**R@10**	**N@50**	**R@50**	**N@10**	**R@10**	**N@50**	**R@50**
Full Agent-R	**0.0845**	**0.1506**	**0.1215**	0.3095	**0.0198**	**0.0354**	**0.0308**	**0.0852**
w/o Memory	0.0825	0.1470	0.1202	0.3095	0.0188	0.0351	0.0300	0.0852
w/o Candidate Adapt.	0.0843	0.1498	0.1191	0.2980	0.0198	0.0354	0.0298	0.0800
w/o Ranking	0.0762	0.1341	0.1163	0.3095	0.0187	0.0330	0.0303	0.0852
w/o Prior Anchor	0.0824	0.1417	0.1209	0.3095	0.0148	0.0285	0.0273	0.0852

alone, especially on Industrial where N@10 decreases from 0.0198 to 0.0148. This indicates that LLM-based refinement is most effective when anchored by existing recommendation intelligence, while memory provides additional contextual preference signals on top of that foundation.

4.3 RQ3: Why Do Heterogeneous Routes Matter?

Figure 2 analyzes whether heterogeneous recommendation routes provide complementary evidence or redundant predictions. The overlap analysis in Figure 2(a) shows that different routes explore distinct candidate regions, with mean off-diagonal Jaccard@50 values of 0.090, 0.068, 0.097, and 0.037 on Software, Industrial, KuaiRec, and MicroLens, respectively. Collaborative routes exhibit the highest overlap, while semantic and memory-based routes provide largely distinct evidence beyond collaborative signals. This diversity translates into substantial recoverable coverage: the union of heterogeneous routes improves Recall@50 over the selected base route by 47.8%, 98.1%, 58.7%, and 95.3% across the four datasets. Figure 2(b) further shows that standalone recommendation strength does not necessarily determine complementary utility. For example, Qwen3-Embedding achieves relatively weak standalone Recall@50 on MicroLens (0.0569) but recovers the largest fraction of base-missed relevant items (3.72%), while Mem-Ret provides additional recovery on Industrial despite weaker standalone performance than collaborative models. These results show that heterogeneous routes should be coordinated based on their ability to expose unresolved preference evidence rather than selected solely by individual ranking quality, motivating the evidence coordination mechanism in Agent-R.

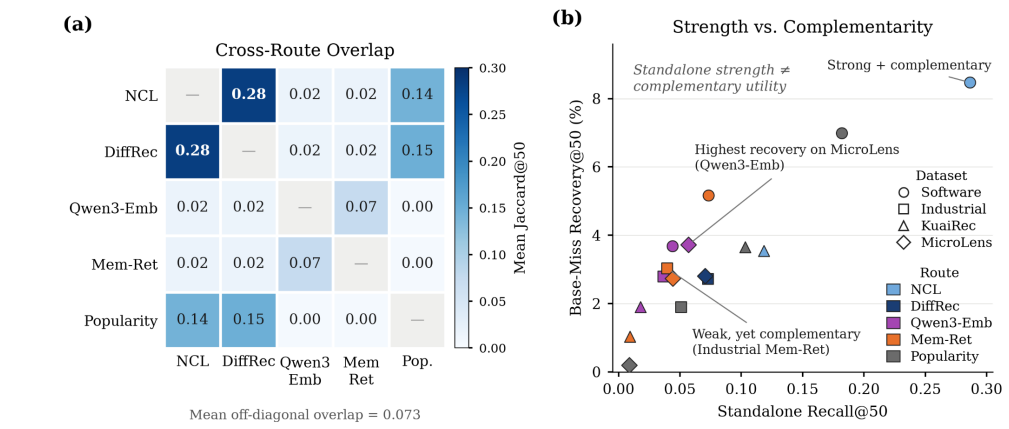

Figure 2: Complementarity analysis of the heterogeneous recommendation routes used by Agent-R. (a) Different routes exhibit low overlap, indicating that they explore distinct candidate regions. (b) Standalone recommendation strength does not necessarily correlate with complementary utility: weaker routes can still recover relevant items missed by the base recommender.

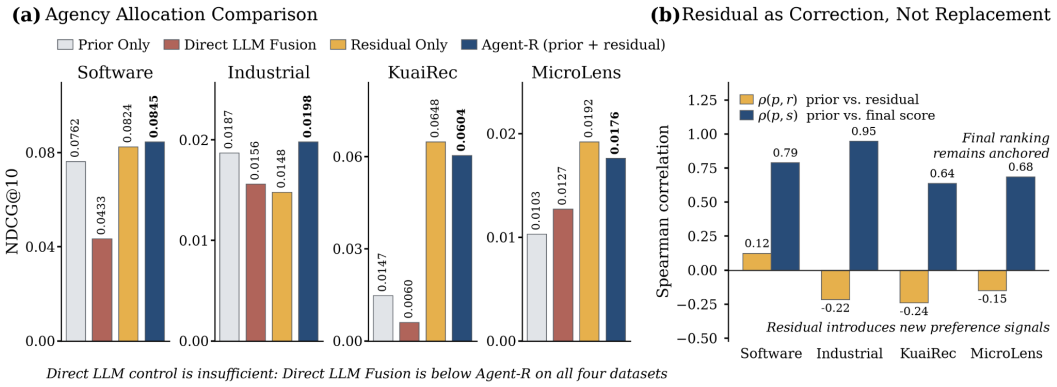

Figure 3: Analysis of LLM agency allocation. (a) Directly assigning the recommendation decision to an LLM using heterogeneous route evidence consistently underperforms structured residual reasoning. (b) The residual introduces preference information that is complementary to the inherited recommendation prior, while the final ranking remains anchored to the original recommendation structure.

4.4 RQ4: Why Does Residual Agency Allocation Matter?

Figure 3 analyzes whether Agent-R benefits from simply introducing LLMs or from allocating LLM agency through structure-preserving residual refinement. As shown in Figure 3(a), Direct LLM Fusion consistently underperforms Agent-R across datasets despite receiving the same heterogeneous recommendation evidence. For example, Direct LLM Fusion reduces NDCG@10 from 0.0845 to 0.0433 on Software and from 0.0604 to 0.0060 on KuaiRec, showing that unrestricted LLM decision making is insufficient to effectively integrate recommendation signals. Figure 3(b) further reveals why residual refinement is effective: the residual has weak correlation with the original recommendation prior ($\rho(p, r)$ ranges from -0.24 to 0.12), indicating that it captures complementary preference information rather than duplicating existing rankings. Meanwhile, the final fused ranking remains strongly correlated with the prior ($\rho(p, s)$ ranges from 0.64 to 0.95), demonstrating that Agent-R preserves reliable recommendation structures while enabling contextual refinement. These results show that effective agentic recommendation does not come from maximizing LLM autonomy, but from allocating LLM agency to unresolved decision components while maintaining existing recommendation intelligence.

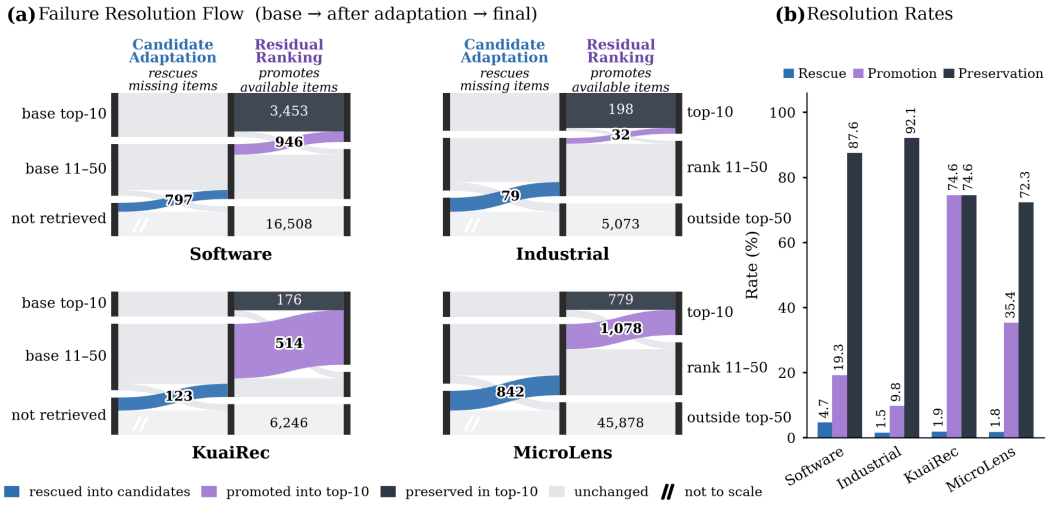

Figure 4: Failure resolution analysis of Agent-R. Candidate adaptation addresses **consideration failures** by rescuing relevant items missing from the base candidate space, while residual ranking addresses **prioritization failures** by promoting available but under-ranked items. The system preserves most already-correct recommendations, showing that the two interventions improve exposure without replacing reliable recommendation structure.

4.5 RQ5: How Does Agent-R Resolve Different Recommendation Failures?

Figure 4 analyzes how the two agency allocation mechanisms in Agent-R address different recommendation failures. The consideration-space mechanism targets cases where relevant items are absent from the base candidate space. Across all four datasets, candidate adaptation consistently improves recall coverage, increasing R@50 by 0.0114, 0.0052, 0.0086, and 0.0099 on Software, Industrial, KuaiRec, and MicroLens, respectively. Despite using only a bounded intervention budget, it successfully introduces previously unreachable relevant items while preserving the majority of existing successful recommendations. In contrast, the preference-space mechanism targets items that are already available but insufficiently prioritized by the original ranking. This effect is especially clear on KuaiRec and MicroLens, where 74.6% and 35.4% of under-ranked relevant items are promoted into the final top-10, respectively. Meanwhile, preservation rates of 87.6% and 92.1% on Software and Industrial show that the refinement does not aggressively disrupt reliable recommendations. These results demonstrate that Agent-R improves recommendation quality through two complementary interventions: expanding the set of items that can be considered and refining the prioritization of items that are already available. More importantly, the analysis validates our agency allocation formulation: different recommendation failures require different forms of LLM intervention rather than a uniform increase of model autonomy.

5 Conclusion

In this paper, we presented **Agent-R**, a structure-preserving multi-agent recommendation framework that formulates the integration of LLMs into recommender systems as an **agency allocation problem** for structured decision integration. Rather than maximizing LLM autonomy or replacing specialized recommendation models, Agent-R allocates LLM reasoning to unresolved decisions through two complementary mechanisms: consideration-space agency allocation for expanding candidate coverage and preference-space residual refinement for improving contextual prioritization. Extensive experiments across diverse recommendation domains demonstrate consistent improvements in both candidate coverage and ranking quality, while further analyses reveal how heterogeneous evidence coordination, residual refinement, and structured memory contribute to these gains. More broadly, our findings suggest that effective agentic recommendation lies not in replacing established recommendation systems with autonomous LLMs, but in carefully allocating LLM agency where additional reasoning provides complementary value.

AI USE STATEMENT

In this work, generative AI tools were used for language polishing, manuscript editing, and improving the clarity of technical descriptions. We did not use generative AI tools to design the experimental methodology, conduct experiments, generate empirical results, or make scientific claims without author verification. All AI-assisted text was reviewed and revised by the authors to ensure correctness, consistency with the implemented system and experimental findings, and appropriate academic presentation. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI tools.

ETHICS STATEMENT

This work studies LLM-based recommendation agents and does not involve human subjects, user studies, or private user data collection. All experiments are conducted on publicly available recommendation benchmarks and offline evaluation protocols. The proposed framework is designed to improve recommendation quality while preserving existing recommendation structures, and we do not introduce additional data collection or user-facing interventions.

REPRODUCIBILITY STATEMENT

We provide the complete method specification in the main paper, including the agency allocation framework, consideration-space and preference-space interventions, and residual learning objective. Additional implementation details, dataset statistics, training configurations, evaluation protocols, and agent prompt templates are provided in the appendix. Anonymous source code and supporting materials for reproducing our experiments are available at `https://anonymous.4open.science/r/agentr_repo-E5E5/`.

REFERENCES

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

Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11 (23-581):81, 2010.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759, 2009.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pp. 191–198, 2016.

Edward A Fox and Joseph A Shaw. Combination of multiple searches. *NIST special publication SP*, 243, 1994.

Aditya Gaydhani, Guangyue Xu, Dhanush Kamath, Ankit Singh, and Alex Li. Unified learning-to-rank for multi-channel retrieval in large-scale e-commerce search. *arXiv preprint arXiv:2602.23530*, 2026.

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

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

Chengkai Huang, Junda Wu, Yu Xia, Zixu Yu, Ruhan Wang, Tong Yu, Ruiyi Zhang, Ryan A Rossi, Branislav Kveton, Dongruo Zhou, et al. Towards agentic recommender systems in the era of multimodal large language models. *arXiv preprint arXiv:2503.16734*, 2025a.

Junjie Huang, Jizheng Chen, Jianghao Lin, Jiarui Qin, Ziming Feng, Weinan Zhang, and Yong Yu. A comprehensive survey on retrieval methods in recommender systems. *ACM Transactions on Information Systems*, 44(1):1–43, 2025b.

Junjie Huang, Jiarui Qin, Jianghao Lin, Ziming Feng, Weinan Zhang, and Yong Yu. Unleashing the potential of multi-channel fusion in retrieval for personalized recommendations. In *Proceedings of the ACM on Web Conference 2025*, pp. 483–494, 2025c.

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

Junguang Jiang, Yanwen Huang, Bin Liu, Xiaoyu Kong, Ziru Xu, Han Zhu, Jian Xu, and Bo Zheng. Large language models are universal recommendation learners. *arXiv e-prints*, pp. arXiv–2502, 2025.

Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1395–1406, 2024.

Yaoyiran Li, Xiang Zhai, Moustafa Alzantot, Keyi Yu, Ivan Vulić, Anna Korhonen, and Mohamed Hammad. Calrec: Contrastive alignment of generative llms for sequential recommendation. In *Proceedings of the 18th ACM conference on recommender systems*, pp. 422–432, 2024.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1785–1795, 2024.

Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Bridging items and language: A transition paradigm for large language model-based recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1816–1826, 2024.

Weiwen Liu, Yunjia Xi, Jiarui Qin, Fei Sun, Bo Chen, Weinan Zhang, Rui Zhang, and Ruiming Tang. Neural re-ranking in multi-stage recommender systems: A review. *arXiv preprint arXiv:2202.06602*, 2022.

Reza Yousefi Maragheh and Yashar Deldjoo. The future is agentic: Definitions, perspectives, and open challenges of multi-agent recommender systems. *arXiv preprint arXiv:2507.02097*, 2025.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Representation learning with large language models for recommendation. In *Proceedings of the ACM web conference 2024*, pp. 3464–3475, 2024.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 14918–14937, 2023.

Alicia Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang, Lichan Hong, Ed H Chi, and Xinyang Yi. Leveraging llm reasoning enhances personalized recommender systems. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13176–13188, 2024.

Lei Wang, Jingsen Zhang, Xiaowen Chen, Yankai Lin, Ruihua Song, Wayne Xin Zhao, and Ji-Rong Wen. Recagent: A novel simulation paradigm for recommender systems. *arXiv preprint arXiv:2306.02552*, 2023.

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

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

Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining*, pp. 806–815, 2024.

Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Information retrieval*, 13(3):254–270, 2010.

Shuyuan Xu, Wenyue Hua, and Yongfeng Zhang. Openp5: An open-source platform for developing, training, and evaluating llm-based recommender systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 386–394, 2024.

Hailan Yang, Zhenyu Qi, Shuchang Liu, Xiaoyu Yang, Xiaobei Wang, Xiang Li, Lantao Hu, Han Li, and Kun Gai. Comprehensive list generation for multi-generator reranking. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2298–2308, 2025.

Shenghao Yang, Weizhi Ma, Peijie Sun, Qingyao Ai, Yiqun Liu, Mingchen Cai, and Min Zhang. Sequential recommendation with latent relations based on large language model. In *Proceedings of the 47th International ACM SIGIR conference on research and development in information retrieval*, pp. 335–344, 2024.

Xiaohan Yu, Li Zhang, Xin Zhao, Yue Wang, and Zhongrui Ma. Ra-rec: An efficient id representation alignment framework for llm-based recommendation. *arXiv preprint arXiv:2402.04527*, 2024.

Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. Llamarec: Two-stage recommendation using large language models for ranking. *arXiv preprint arXiv:2311.02089*, 2023.

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

Jiang Zhang, Sumit Kumar, Wei Chang, Yubo Wang, Feng Zhang, Weize Mao, Hanchao Yu, Aashu Singh, Min Li, and Qifan Wang. Optimizing recall or relevance? a multi-task multi-head approach for item-to-item retrieval in recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5194–5204, 2025.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Xin Zhao, Leyu Lin, and Ji-Rong Wen. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *The Web Conference 2024*, 2024b.

Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. Collaborative large language model for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pp. 3162–3172, 2024.

APPENDIX

A RELATED WORK

A.1 LLM-BASED AGENTIC RECOMMENDATION

Large language models have recently become an important direction for advancing recommender systems by providing semantic understanding, instruction following, and natural-language interaction capabilities. Early studies mainly explore LLMs as recommendation models by reformulating recommendation tasks as language generation or instruction following problems. P5 Geng et al. (2022) and OpenP5 Xu et al. (2024) unify diverse recommendation tasks under a text-to-text learning paradigm, while TALLRec Bao et al. (2023), LLM4Rec Zhu et al. (2024), LLaRA Liao et al. (2024), and related approaches investigate adapting LLMs to sequential recommendation, personalization, and ranking tasks Yang et al. (2024); Lin et al. (2024). Beyond directly using LLMs as recommenders, another line of work leverages LLMs to enhance conventional recommendation models through semantic augmentation, user profile construction, and representation alignment Yu et al. (2024); Li et al. (2024). Methods such as LLMRec Wei et al. (2024), RLMRec Ren et al. (2024), and LLM-enhanced collaborative filtering approaches Kim et al. (2024) incorporate language-derived knowledge into collaborative and sequential recommendation models, demonstrating the potential of LLMs for enriching existing recommendation signals. Recent studies further explore LLM reasoning and generalization capabilities for personalized recommendation Tsai et al. (2024); Jiang et al. (2025).

More recently, research has moved toward **agentic recommendation**, where LLMs are equipped with planning, memory, interaction, and tool-use capabilities to support more adaptive recommendation processes. RecAgent Wang et al. (2023), InteRecAgent Huang et al. (2025d), Agent4Rec Zhang et al. (2024a), and related systems explore LLM-based recommendation agents that can interact with users, maintain contextual information, and perform multi-step recommendation tasks. Multi-agent frameworks such as AgentCF Zhang et al. (2024b) and MACRec Wang et al. (2024b) further investigate collaborative reasoning among multiple recommendation agents, while recent studies discuss broader agentic recommendation paradigms and open challenges Huang et al. (2025a); Maragheh & Deldjoo (2025). Memory-oriented and interactive recommendation agents further explore how long-term user information and adaptive interaction mechanisms can improve personalization Wang et al. (2024a). These studies demonstrate the potential of LLM agents for flexible and interactive recommendation. However, existing approaches primarily focus on improving LLM capability or increasing agent autonomy, while overlooking *how LLM decision authority should be allocated within recommendation pipelines that already contain specialized intelligence*. Agent-R addresses this gap by studying agency allocation and assigning LLM reasoning to unresolved recommendation decisions while preserving the structures learned by existing recommenders.

A.2 MULTI-ROUTE FUSION FOR RECOMMENDATION

Modern recommender systems often combine multiple retrieval routes or ranking experts to exploit complementary inductive biases across heterogeneous recommendation models. Traditional fusion approaches aggregate outputs from multiple systems through score- or rank-based strategies, including score fusion methods such as CombSUM and CombMNZ (Fox & Shaw, 1994) and rank aggregation methods such as Reciprocal Rank Fusion (RRF) (Cormack et al., 2009). Learning-based fusion approaches further treat route outputs as ranking features and optimize their combination through supervised objectives, with learning-to-rank methods such as LambdaMART (Burges, 2010; Wu et al., 2010) widely used for combining heterogeneous signals. Recent recommender systems further adopt multi-stage architectures where multiple candidate generators provide complementary retrieval signals before downstream ranking (Covington et al., 2016). More recent studies investigate multi-channel retrieval fusion through global or personalized channel weighting, query-dependent fusion strategies, and reranking over multiple list generators (Huang et al., 2025b; Zhang et al., 2025; Huang et al., 2025c; Gaydhani et al., 2026; Yang et al., 2025). These approaches demonstrate the value of heterogeneous recommendation routes, but mainly focus on learning better aggregation functions or fusion weights over route outputs.

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

Dataset	Users	Items	Interactions	Density
Software	25,366	12,806	404,826	0.125%
Industrial	5,615	17,630	79,442	0.080%
DoubanBook	1,777	70,205	155,525	0.125%
KuaiRec	7,171	10,611	846,408	1.112%
MicroLens	50,000	19,220	359,708	0.037%

With the emergence of LLMs, recent studies further explore using LLMs as fusion or ranking modules over retrieved candidates. Existing approaches investigate LLM-based recommendation judgment (Bao et al., 2023), zero-shot ranking (Hou et al., 2024), listwise reranking (Sun et al., 2023), and LLM-based recommendation ranking frameworks (Yue et al., 2023). While these methods leverage the reasoning ability of LLMs, they typically assign direct decision authority to the model and may overlook the specialized recommendation structures captured by existing experts. In contrast, Agent-R studies how LLM agency should be allocated within a multi-route recommendation pipeline, where LLM agents coordinate heterogeneous evidence and refine unresolved preference decisions while preserving existing recommendation intelligence.

B Experiments

Beyond the main results, we provide additional analyses to further understand the effectiveness and behavior of Agent-R. We first analyze why heterogeneous recommendation routes provide complementary evidence and how Agent-R benefits from allocating LLM agency across different decision spaces. We then study how Agent-R resolves different recommendation failures through consideration-space and preference-space interventions, followed by analyses of memory contribution, residual preference refinement, route robustness, and computational efficiency. Together, these experiments provide a deeper understanding of how Agent-R integrates LLM reasoning with existing recommendation intelligence while preserving reliable recommendation structures.

B.1 Experimental Setup

Datasets. We evaluate Agent-R on five representative datasets covering diverse recommendation domains, including Software, Industrial, KuaiRec, MicroLens, and DoubanBook. Software and Industrial are derived from Amazon review datasets, while KuaiRec, MicroLens, and DoubanBook represent domain-generalization scenarios. Table 3 summarizes the dataset statistics. All datasets follow chronological splits, where interactions are ordered by timestamp and future interactions are reserved for validation and testing.

Recommendation Routes and Baselines. Agent-R builds upon a heterogeneous recommendation route bank consisting of collaborative, semantic, memory-based, and popularity-based routes. Specifically, we use NCL and DiffRec as collaborative recommenders, Qwen3-Embedding-4B as semantic retrieval, Mem-Ret as memory retrieval, and popularity as a global preference prior. Each route produces a ranked candidate list, and the base collaborative route is selected according to validation Recall@50.

We compare Agent-R with specialized recommendation models and multi-route fusion baselines. The fusion baselines receive the same route outputs as Agent-R, including RRF, CombMNZ, validation-weighted fusion, Logistic fusion, and LambdaMART. We additionally compare with Direct LLM Fusion, where Qwen2.5-7B-Instruct directly ranks candidates without the structure-preserving residual mechanism.

Agent-R Implementation. Agent-R uses Qwen2.5-7B-Instruct as the backbone model for all agents. The Memory Module constructs structured STM/LTM user representations from interaction histories, including recent intents, long-term categories, preferred brands, and stable preferences. The Planning Agent performs consideration-space agency allocation by generating bounded evidence coordination policies over heterogeneous routes. The planner outputs constrained source adjustments, which are mapped to bounded channel weights by the deterministic executor.

Table 4: Oracle coverage of heterogeneous route union compared with the selected base route. The union is an upper bound on available evidence rather than a budget-matched recommendation method.

Dataset	Base	Base R@50	Union R@50	Gain
Software	DiffRec	0.2980	0.4406	+47.8%
Industrial	NCL	0.0800	0.1584	+98.1%
KuaiRec	DiffRec	0.1203	0.1910	+58.7%
MicroLens	NCL	0.0726	0.1418	+95.3%

The Ranking Agent performs preference-space residual refinement over the adapted candidate set. For each user–item pair, the final ranking score is:

$$s_u(i) = p_u(i) + \alpha r_\theta(u, i),$$

where $p_u(i)$ is the inherited recommendation prior and $r_\theta(u, i)$ is the LLM residual score. The residual score is bounded through a $\tanh$ projection to constrain LLM intervention.

Training Details. The residual Ranking Agent is trained with Qwen2.5-7B-Instruct using LoRA adaptation. The LoRA configuration uses rank $r = 16$, scaling factor $\alpha = 32$, and dropout 0.05. The residual prediction head is fully optimized together with the LoRA parameters.

We optimize the fused recommendation score using a listwise ranking objective:

$$\mathcal{L} = \mathcal{L}_{rank} + \beta \mathcal{L}_{reg},$$

where $\mathcal{L}_{rank}$ optimizes the final score $p_u(i) + \alpha r_\theta(u, i)$ and $\mathcal{L}_{reg}$ regularizes residual magnitude. The optimizer is AdamW with learning rates 1×10^{-4} for LoRA parameters and 1×10^{-3} for the prediction head, with gradient clipping of 1.0. Training uses bf16 precision, FlashAttention-2, and gradient checkpointing. The effective batch size is 64 users per optimization step using 8 GPUs with gradient accumulation. The residual strength α and recommendation prior type are selected on the validation set.

Evaluation Protocol. We evaluate all methods using Recall@10, Recall@50, NDCG@10, and NDCG@50. All model selection decisions, including route selection, prior selection, planner policies, checkpoints, and residual strength, are determined using validation data only. The final recommendation list contains 50 items, and all reported results are computed over the complete test population. Statistical significance is measured using paired bootstrap testing over users with 10,000 resamples.

Prompt Templates. For reproducibility and readability, we provide the normalized prompt templates used by the Preference Memory Module, Planning Agent, and Ranking Agent in Appendix B.9.

B.2 Why Do Heterogeneous Routes Matter?

Table 4 further quantifies the complementary evidence available from heterogeneous recommendation routes by comparing the selected base route with the oracle union of all available routes. Although the base recommender already provides a strong recommendation backbone, combining the candidate spaces exposed by different routes substantially enlarges the available relevant-item coverage. The oracle union improves Recall@50 by 47.8% on Software, 98.1% on Industrial, 58.7% on KuaiRec, and 95.3% on MicroLens over the corresponding base routes. This measures the upper bound of available evidence rather than the achieved coverage. The gains are especially notable on Industrial and MicroLens, where complementary routes provide substantial additional candidate coverage beyond the existing backbone. These results do not imply that the oracle union itself is a practical recommendation strategy, since it is not constrained by the final candidate budget; instead, they quantify the amount of complementary recommendation evidence available across heterogeneous routes. This observation motivates Agent-R's consideration-space agency allocation, where the Planning Agent coordinates heterogeneous route evidence to expose useful complementary candidates while preserving the reliable recommendation backbone.

Table 5: Comparison of different LLM agency allocation strategies. Direct LLM Fusion allows unrestricted LLM decision making, while Agent-R applies LLM reasoning as a residual correction over an explicit recommendation prior.

Dataset	Method	N@10	R@10	N@50	R@50
Software	Prior Only	0.0762	0.1341	0.1163	0.3095
	Residual Only	0.0824	0.1417	0.1209	0.3095
	Direct LLM Fusion	0.0433	0.0810	0.0757	0.2187
	Agent-R	**0.0845**	**0.1506**	**0.1215**	0.3095
Industrial	Prior Only	0.0187	0.0330	0.0303	0.0852
	Residual Only	0.0148	0.0285	0.0273	0.0852
	Direct LLM Fusion	0.0156	0.0265	0.0258	0.0698
	Agent-R	**0.0198**	**0.0354**	**0.0308**	0.0852
KuaiRec	Prior Only	0.0147	0.0329	0.0349	0.1290
	Residual Only	**0.0648**	0.0961	**0.0721**	0.1290
	Direct LLM Fusion	0.0060	0.0141	0.0166	0.0643
	Agent-R	0.0604	**0.0962**	0.0677	0.1290
MicroLens	Prior Only	0.0103	0.0215	0.0232	0.0824
	Residual Only	**0.0192**	**0.0389**	**0.0288**	0.0824
	Direct LLM Fusion	0.0127	0.0268	0.0197	0.0572
	Agent-R	0.0176	0.0371	0.0276	0.0824

Table 6: Failure resolution statistics of Agent-R.

Dataset	Rescue Rate	Promotion Rate	Preservation Rate	ΔR@50
Software	4.74%	19.25%	87.55%	+0.0114
Industrial	1.55%	9.79%	92.09%	+0.0052
KuaiRec	1.95%	74.60%	74.58%	+0.0086
MicroLens	1.82%	35.40%	72.33%	+0.0099

B.3 Why Does Residual Agency Allocation Matter?

Table 5 provides a detailed comparison of alternative LLM agency allocation strategies. Direct LLM Fusion consistently underperforms Agent-R, showing that directly assigning unrestricted decision authority to the LLM over heterogeneous recommendation evidence is insufficient without structure-preserving constraints. Removing the recommendation prior produces mixed results: it can improve performance in domains where the inherited ranking is less reliable, such as KuaiRec and MicroLens, but consistently hurts stronger-prior settings such as Software and Industrial. These observations show that the recommendation prior is not merely an additional signal, but a structural anchor that allows the LLM residual to refine unresolved preference decisions without discarding reliable recommendation intelligence.

B.4 How Does Agent-R Resolve Different Recommendation Failures?

Table 6 provides a detailed breakdown of how Agent-R improves recommendation quality through different forms of agency allocation. The consideration-space mechanism addresses candidate exposure failures by recovering relevant items that are absent from the base candidate space. Across all datasets, candidate adaptation improves recall coverage, as reflected by the positive R@50 gains. The rescue rate remains intentionally moderate because the intervention budget is bounded and only a limited portion of the candidate space is allocated to complementary evidence. In contrast, the preference-space mechanism addresses prioritization failures where relevant items are already available but insufficiently ranked. This effect is reflected by the promotion rate, especially on KuaiRec and MicroLens. Meanwhile, the preservation rates on Software and Industrial demonstrate that residual refinement maintains reliable recommendation structures rather than replacing the original ranking. These results further validate the separation between consideration-space and preference-space agency allocation in Agent-R.

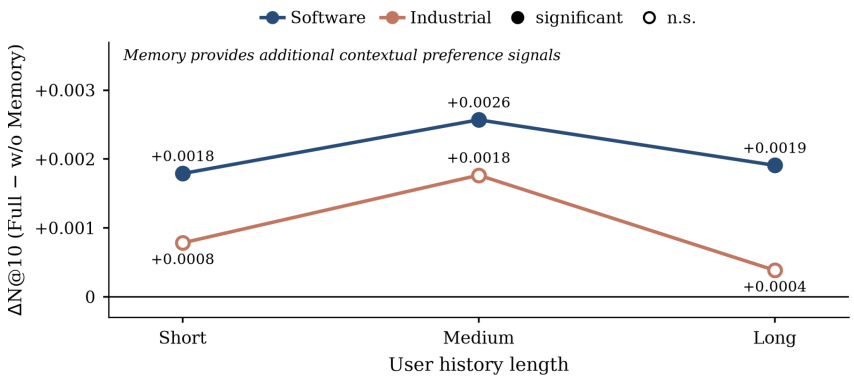

Figure 5: Memory contribution across user history lengths. The gain is measured as the difference between Agent-R with LLM STM/LTM memory and the deterministic user profile. Memory provides additional contextual preference signals across different history regimes, with the largest improvement appearing for medium-history users.

B.5 When Does Memory Help?

We further analyze the contribution of LLM-based STM/LTM memory within the preference-space agency allocation process. Figure 5 shows the performance gain of Agent-R with memory across different user history lengths on representative domains. Memory consistently improves recommendation quality across all history regimes on Software, with NDCG@10 gains of +0.0018, +0.0026, and +0.0019 for short-, medium-, and long-history users, respectively. The improvement is most pronounced for medium-history users, suggesting that memory is particularly useful when users provide sufficient interaction signals but their contextual preferences cannot be fully summarized by deterministic statistics. A similar positive trend is observed on Industrial, where memory improves all history groups, with the largest gain appearing in the medium-history regime. These results indicate that LLM memory provides complementary contextual preference representations for the Ranking Agent rather than simply extending the observable interaction history.

Table **??** further summarizes the impact of LLM memory. Replacing STM/LTM memory with a deterministic profile consistently reduces performance on the representative domains, with a statistically significant degradation on Software. Although the improvement is smaller than the gains from consideration-space adaptation and residual preference refinement, memory provides additional contextual information that complements the recommendation evidence already captured by specialized routes. The history-based analysis further shows that the benefit comes from contextual preference abstraction rather than simply increasing the amount of user interaction information.

B.6 Does the Residual Ranker Correct or Replace the Recommendation Prior?

Figure 6 analyzes whether the Ranking Agent acts as a replacement for the existing recommendation structure or as a bounded preference-space residual refinement mechanism. First, Figure 6(a) shows that the learned residual captures preference information that is largely complementary to the inherited recommendation prior. The correlation between the prior score and residual score remains close to zero across all datasets ($\rho(p, r) \in [-0.24, 0.12]$), indicating that the residual does not simply reproduce the ranking behavior already encoded by specialized recommenders. At the same time, the final ranking remains strongly correlated with the prior ($\rho(p, s) \in [0.64, 0.95]$), demonstrating that the Ranking Agent preserves reliable recommendation structures while refining unresolved preference decisions.

Figure 6(b) further verifies that the residual signal is aligned with recommendation relevance rather than being an arbitrary perturbation. Across all datasets, the ground-truth item appears in the top residual decile substantially more frequently than random expectation (10%), reaching 32.7% on Software, 18.5% on Industrial, 61.5% on KuaiRec, and 29.3% on MicroLens. These results show that the residual selectively identifies candidates whose contextual preference information is underrepresented by the original ranking. Together, the analysis supports the design of Agent-R: LLM

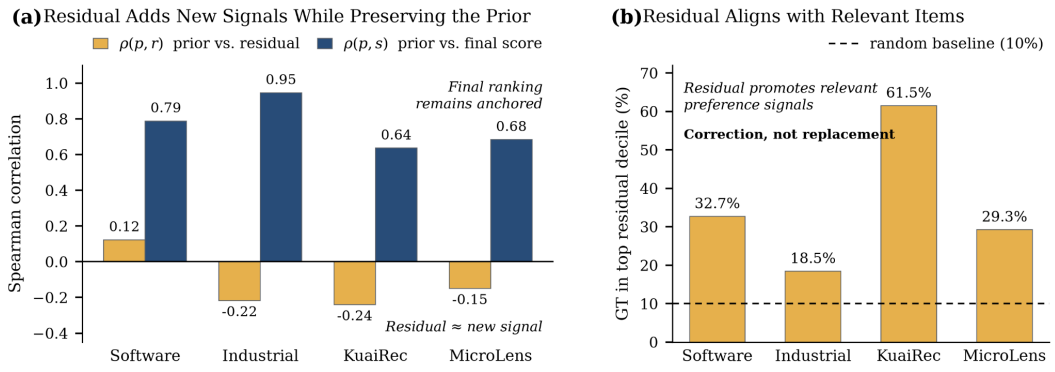

Figure 6: Residual ranking behavior analysis. (a) The residual score introduces preference signals that are weakly correlated with the inherited recommendation prior, while the final fused ranking remains strongly anchored to the prior. (b) Candidates receiving high residual scores are substantially more likely to contain the ground-truth item than random candidates, indicating that the residual performs targeted preference correction rather than arbitrary perturbation.

Table 7: Residual behavior statistics. The residual should provide complementary preference information while maintaining alignment between the final ranking and the original recommendation prior.

Dataset	$\rho(p, r)$	$\rho(p, s)$	GT in residual top decile
Software	0.123	0.788	32.7%
Industrial	-0.217	0.946	18.5%
KuaiRec	-0.241	0.636	61.5%
MicroLens	-0.150	0.684	29.3%

reasoning should be introduced as bounded residual agency that contributes additional preference information while preserving the recommendation intelligence encoded by specialized models.

Table 7 provides additional quantitative evidence for the residual correction mechanism. The low prior–residual correlations demonstrate that the LLM residual contains information beyond the existing recommender ordering, while the high prior–final correlations confirm that the residual operates as a controlled refinement rather than an unconstrained reranking function. Furthermore, the strong enrichment of ground-truth items among high-residual candidates shows that the learned correction is preference-aware: candidates promoted by the residual are substantially more likely to correspond to actual user interests than candidates selected at random. These observations validate the role of the Ranking Agent as a residual preference model that complements, rather than replaces, specialized recommendation structure.

B.7 How Robust Is Agent-R to Route Composition?

Figure 7 analyzes the contribution and robustness of individual recommendation routes under leave-one-route-out evaluation. Removing collaborative routes consistently causes the largest degradation, demonstrating that Agent-R preserves and builds upon strong collaborative recommendation structures rather than replacing them. For example, removing DiffRec on Software decreases R@50 by 0.0286, while removing either NCL or DiffRec on KuaiRec reduces R@50 by 0.0149 and 0.0185, respectively. These results confirm that collaborative routes provide the primary recommendation backbone.

At the same time, auxiliary routes provide complementary and domain-dependent evidence for consideration-space agency allocation. Popularity contributes substantially on Software and KuaiRec, where removing it decreases R@50 by 0.0104 and 0.0130, respectively. Semantic and memory routes become particularly useful on MicroLens, where removing Qwen3-Embedding and Mem-Ret reduces R@50 by 0.0079 and 0.0061. In contrast, some auxiliary routes have limited

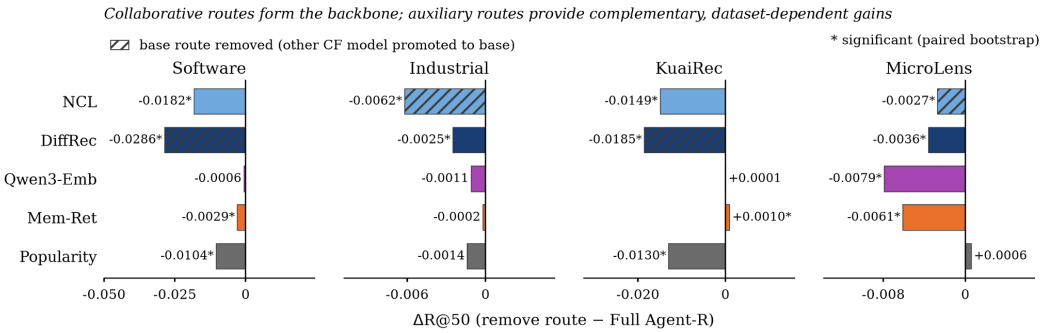

Figure 7: Route robustness analysis under leave-one-route-out evaluation. The performance degradation after removing individual routes shows that collaborative routes provide the primary recommendation backbone, while auxiliary semantic, memory, and popularity routes contribute complementary dataset-dependent evidence. Hatched bars indicate removal of the base collaborative route, where the remaining collaborative model is promoted to the base slot.

Table 8: Leave-one-route-out robustness analysis. Values report the change in Recall@50 after removing one route compared with the full Agent-R system.

Dataset	-NCL	-DiffRec	-Qwen3-Emb	-Mem-Ret	-Popularity
Software	-0.0182*	-0.0286*	-0.0006	-0.0029*	-0.0104*
Industrial	-0.0062*	-0.0025*	-0.0011	-0.0002	-0.0014
KuaiRec	-0.0149*	-0.0185*	+0.0001	+0.0010*	-0.0130*
MicroLens	-0.0027*	-0.0036*	-0.0079*	-0.0061*	+0.0006

impact in specific domains, highlighting that their value comes from exposing complementary evidence rather than uniformly improving ranking quality. These results are consistent with the route complementarity analysis in RQ3 and show why Agent-R benefits from coordinating heterogeneous recommendation routes instead of relying on a single recommendation source.

Table 8 provides a detailed view of route-level contribution. The strongest and most consistent effects come from collaborative routes, whose removal substantially decreases candidate quality across datasets. Meanwhile, auxiliary routes exhibit complementary behavior: their contribution depends on the underlying domain, with semantic and memory routes becoming more important in MicroLens and popularity providing additional value in Software and KuaiRec. This confirms that Agent-R does not rely on a single dominant retrieval source; instead, it exploits heterogeneous routes whose individual utility varies according to the recommendation environment.

B.8 What Is the Computational Cost of Agent-R?

Figure 8 analyzes the computational cost of Agent-R and how LLM reasoning is allocated across different recommendation decisions. Unlike approaches that invoke LLMs directly for unrestricted recommendation decisions, Agent-R separates reusable contextual representation and evidence coordination from online preference refinement. The Memory Module and Planning Agent operate offline and can be cached for each user, while the Executor performs deterministic CPU-based candidate adaptation. As shown in Figure 8(a), the dominant online cost comes from the Ranking Agent, which performs bounded preference refinement over the 50-item candidate slate rather than reasoning over the full catalog. This design concentrates LLM computation on the decision stage where additional contextual preference reasoning provides the most complementary value.

Figure 8(b) further compares the allocation of LLM reasoning with Direct LLM Fusion. Direct LLM Fusion spends its budget on repeatedly asking the LLM to make independent candidate-level decisions, requiring approximately 41–46 pointwise calls per user. In contrast, Agent-R distributes LLM reasoning across cached memory construction, bounded evidence coordination, and structured residual ranking. Although Agent-R does not aim to minimize LLM computation, its reasoning budget is allocated within a constrained recommendation pipeline rather than replacing the recommendation

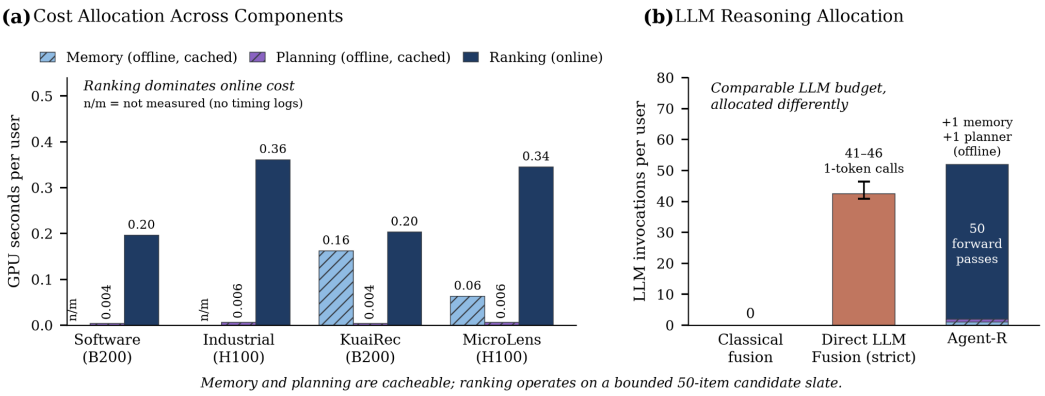

Figure 8: Efficiency analysis of Agent-R. (a) The online computation is dominated by the bounded candidate-level Ranking Agent, while memory and planning are offline and cacheable. (b) Agent-R allocates a comparable amount of LLM reasoning budget differently from direct LLM fusion: reasoning is concentrated on structured residual refinement rather than unrestricted candidate decision making.

Table 9: Serving-time cost breakdown of Agent-R. Memory and Planning are offline/cacheable, while Ranking performs online bounded candidate refinement.

Component	Online	Calls / User	Main Cost
Memory Module	No	1	Offline STM/LTM generation
Planning Agent	No	1	Offline policy generation
Executor	Yes	0	CPU deterministic scoring
Ranking Agent	Yes	50 forwards	Candidate-level cross-encoding

system with unconstrained LLM inference. This analysis highlights the practical principle behind Agent-R: the goal is not to maximize LLM usage, but to allocate LLM reasoning to the decisions where it provides complementary value.

Table 9 summarizes the computational structure of Agent-R. The two LLM components responsible for user representation and candidate-space coordination are executed offline and can be reused until the user's preference state changes. The only online neural component is the Ranking Agent, which operates on a fixed 50-item candidate slate. Therefore, the computational cost of Agent-R grows with the bounded recommendation slate rather than the size of the entire item catalog. This design enables Agent-R to incorporate LLM reasoning while maintaining a practical deployment structure.

Additional measurements show that the Ranking Agent requires approximately 9k–11k prompt tokens per user and dominates online GPU cost, whereas the Planning Agent requires only around 520 prompt tokens and 36–40 output tokens per user. Compared with strict Direct LLM Fusion, which requires approximately 41–46 LLM calls per user, Agent-R allocates similar reasoning resources through structured components rather than unrestricted candidate-level decisions.

B.9 Representative Agent Prompts

We provide representative prompt templates for the three LLM-based modules in Agent-R: the Memory Module, Planning Agent, and Ranking Agent. The Memory Module constructs structured user preference states from historical interactions, the Planning Agent determines bounded evidence coordination policies over heterogeneous routes, and the Ranking Agent estimates contextual preference residuals over an existing recommendation prior. For readability, we provide cleaned and normalized prompt templates derived from the prompts used in our experiments. We omit implementation-only fields and runtime bookkeeping while retaining the core inputs, decision rules, and output structures that determine each module's behavior.

Memory Module Prompt

```
You are a user Memory Module for a recommender system. You convert
interaction evidence into compact short-term and long-term user
memory for personalized recommendation. Use only the provided
evidence and do not invent unsupported preferences.

RUNTIME INPUTS

USER HISTORY SUMMARY:
<interaction statistics and history length>

RECENT INTERACTIONS:
<recent items with title, brand, and category information>

USER STATISTICS:
<frequent categories, brands, and representative items>

INSTRUCTIONS

Construct reusable preference memories from the observed evidence.

Identify:   -- short-term interests and recent intents;
-- stable long-term categories and brands;
-- persistent preference patterns;
-- representative historical items.

Keep each memory field concise and suitable for downstream
recommendation. Do not infer preferences that are not supported
by the provided interactions.

OUTPUT

Return a JSON object containing:
{  "short_term_intents": [...],
"long_term_categories": [...],
"long_term_brands": [...],
"stable_preferences": [...],
"representative_items": [...]
}
Output strict JSON only.
```

Planning Agent Prompt

```
You are the Coordination Planner in a multi-route recommender
system. You do not select recommendation items. Instead, you
define a constrained intervention policy that a deterministic
executor applies over heterogeneous recommendation routes.

RUNTIME INPUTS

USER COORDINATION DIAGNOSTICS:
<route reliability, validation statistics, cross-route support, and
user preference compatibility>

AVAILABLE ROUTES:
<collaborative, semantic, memory, and popularity routes>

INSTRUCTIONS
```

```
Determine how much each evidence source should contribute for this
user.
Consider:    -- route reliability and validation strength;
-- agreement among multiple routes;
-- compatibility between retrieved evidence and user preference.

You are not responsible for final item selection.  Only output
bounded evidence adjustments:
  down:  reduce contribution;
neutral:  keep the default contribution;
up:  emphasize reliable complementary evidence.

Prefer conservative adjustments and avoid unnecessary intervention.

OUTPUT

Return a JSON policy:
{   "source_trust":  {
"semantic":  "...",
"memory":  "...",
"collaborative":  "...",
"popularity":  "..."
},
"evidence_weights":  {...  }
}
Output strict JSON only.
```

Ranking Agent Prompt

```
You are a recommendation ranking expert judging one candidate
for a user.  Your goal is to estimate candidate-level preference
compatibility using user context and recommendation evidence.

RUNTIME INPUTS

USER PROFILE:
<structured short-term and long-term preference memory>

CANDIDATE ITEM:
<title, brand, and description>

RECOMMENDATION EVIDENCE:
<base rank, candidate provenance, and route-level evidence>

INSTRUCTIONS

Assess how likely this candidate matches the user's next
interaction.
Use both user preference context and recommendation evidence.
Provide complementary preference information beyond the
existing recommendation prior rather than replacing the original
recommendation structure.

OUTPUT

A contextual representation used to compute a bounded residual
preference signal for final ranking.  No text generation is
required.
```