Internalizing Multi-Agent Reasoning for Accurate and Efficient LLM-based Recommendation

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are reshaping recommender systems by leveraging extensive world knowledge and semantic reasoning to interpret user intent. However, effectively integrating these capabilities with collaborative signals while avoiding prohibitive inference latency remains a critical bottleneck. To address this, we propose a trajectory-driven internalization framework to develop a Single-agent Trajectory-Aligned Recommender (STAR). Specifically, to internalize complex reasoning capabilities into a single efficient model, we first design a multi-agent teacher system capable of multi-turn tool usage and reflection. This teacher utilizes a Collaborative Signal Translation mechanism to explicitly convert latent behavioral patterns into descriptive natural language evidence to enhance reasoning accuracy. Subsequently, a trajectory-driven distillation pipeline transfers this agentic logic, including planning, tool usage, and self-reflection, into the compact STAR model. Extensive experiments demonstrate that STAR surpasses its teacher by 8.7% to 39.5% while eliminating iterative latency, paving the way for real-time, reasoning-enhanced recommendation.

1 Introduction

Large Language Models (LLMs) (Achiam et al., 2023; Guo et al., 2025) are reshaping recommender systems, moving beyond rigid ID-based matching toward semantic understanding and natural-language generation (Lin et al., 2025; Peng et al., 2025; Zhu et al., 2025a). Yet effective recommendation heavily relies on collaborative signals, defined as the behavioral consensus distilled from massive user interactions such as item co-occurrence and community-level preference regularities. When operating primarily on textual descriptions, LLMs do not naturally observe such signals (Zhang et al., 2024b; Liu et al., 2025), creating a disconnect between semantic capabilities and behavioral patterns. Therefore, bridging this gap by leveraging collaborative information is crucial for improving LLM-based recommendation.

To fuse collaborative signals with semantic features, a dominant paradigm integrates LLMs as representation enhancers (Liu et al., 2024; He et al., 2025) to augment collaborative filtering with semantic embeddings. While this approach effectively injects collaborative signals, it reduces the LLM to a feature extractor. By compressing complex intents into opaque vectors, it inherently sacrifices the model's core advantage of explicit, step-by-step reasoning, rendering the recommendation process uninterpretable.

Alternatively, the second paradigm employs tool-augmented agents (Wang et al., 2024b; Wu et al., 2024; Zhu et al., 2025b; Tang et al., 2025; Xia et al., 2025) to actively retrieve information. This approach directly leverages the reasoning power of LLMs to query external data. Yet, this direction faces two critical limitations. First, most agents rely on either opaque numerical scores that hide the rationale or raw item metadata that lacks behavioral consensus, failing to verbalize the statistical reasons behind user interactions. Second, their iterative multi-turn nature causes latency to scale linearly with reasoning depth, rendering them impractical for real-time serving.

Therefore, enabling explicit reasoning over collaborative signals while maintaining real-time efficiency is imperative for advancing LLM-based recommender systems. To tackle these challenges, we propose a strategy to internalize the intelligence of multi-agent systems. By distilling the agentic workflow into a unified model, we retain rigorous inference capabilities while eliminating the coordination overhead during inference. We realize this via a two-phase framework. First, we construct the Multi-Agent Recommender System (MARS) as a training-time teacher. MARS features a novel Collaborative Signal Translation mechanism that

explicitly retrieves behavioral neighbors from the user-item graph and verbalizes them into natural language evidence, enabling explicit deduction over collaborative signals. Second, we introduce a trajectory-driven distillation pipeline that synergizes Supervised Fine-Tuning (SFT) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to transfer sophisticated agentic skills into a single model. This alignment stage incentivizes the student to capture the agentic decision-making logic by learning specifically when to invoke tools and how to self-reflect within a single generation pass. Extensive experiments demonstrate that the resulting STAR (Single-agent Trajectory-Aligned Recommender) achieves superior performance, surpassing its teacher by 8.7%–39.5% across various scenarios with significantly reduced inference latency and minimal resource consumption.

Our contributions are threefold:

- We propose a Collaborative Signal Translation mechanism within MARS, which bridges the gap by verbalizing latent behavioral patterns into concise evidence, enabling LLMs to effectively reason over collaborative signals.
- We introduce a trajectory-driven distillation pipeline to transfer sophisticated agentic skills including tool use and self-reflection into a single model.
- We deliver STAR, a Single-agent Trajectory-Aligned Recommender that achieves superior performance, surpassing its teacher by 8.7%–39.5% while substantially reducing inference latency and hardware requirements.

2 Related Works

LLMs as Representation Enhancers. To fuse semantic and collaborative signals, dominant approaches integrate LLMs as representation enhancers. Prior works employ instruction tuning to align representations (Bao et al., 2023), while recent methods inject collaborative embeddings via soft prompts or efficient fine-tuning (Zhu et al., 2024; Liu et al., 2024; Wu et al., 2024; Ren et al., 2024; Zhang et al., 2025). While effective, these methods compress complex user-item interaction graphs into opaque vectors. This compression inherently sacrifices reasoning transparency, preventing the model from explicitly verbalizing the behavioral consensus underlying its predictions.

Agentic Recommender Systems. Unlike representation enhancers that rely on static vectors, agentic frameworks adopt a Plan-Execute paradigm to actively retrieve external information (Wang et al., 2024b,a; Xia et al., 2025; Huang et al., 2025; Ma et al., 2025). Notably, the Interactive Recommendation Agent (Tang et al., 2025) incorporate active commands but rely on general instruction tuning, failing to fully internalize complex tool usage and self-reflection. Since standard distillation also misses such dynamic logic, STAR addresses this by internalizing agentic capabilities into a single efficient model via trajectory-driven distillation.

3 Methodology

In this section, we propose a framework to internalize the explicit reasoning capabilities of multi-agent systems, specifically active tool use and self-reflection, into a compact unified model. As illustrated in Figure 1, the methodology begins with the Multi-Agent Recommender System (MARS), which synthesizes reasoning-rich interaction trajectories to explicitly verbalize latent collaborative signals. These agentic skills are subsequently distilled into the Single-agent Trajectory-Aligned Recommender (STAR) via a trajectory-driven pipeline. This paradigm enables the student model to retain the reasoning depth of the teacher while eliminating inference-time coordination overhead.

3.1 Task Formulation

Let $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items. For a specific user $u \in \mathcal{U}$, the interaction history is defined as a chronological sequence $s_u = [v_1, v_2, \ldots, v_t]$ where $v_i \in \mathcal{I}$ represents the item interacted with at step i.

In the paradigm of LLM-based recommendation, we formulate this task as a conditional text generation problem. The interaction history s_u is converted into a natural language instruction x using a predefined prompt template. The objective is to learn a policy π_θ that maximizes the probability of generating a response o conditioned on x:

$$p(o|s_u) = \pi_\theta(o|x). \tag{1}$$

Here the generated sequence o represents the textual output containing the ranked list of candidate items. This formulation treats recommendation as a sequence generation task and aims to rank the ground-truth next item v_{t+1} as highly as possible.

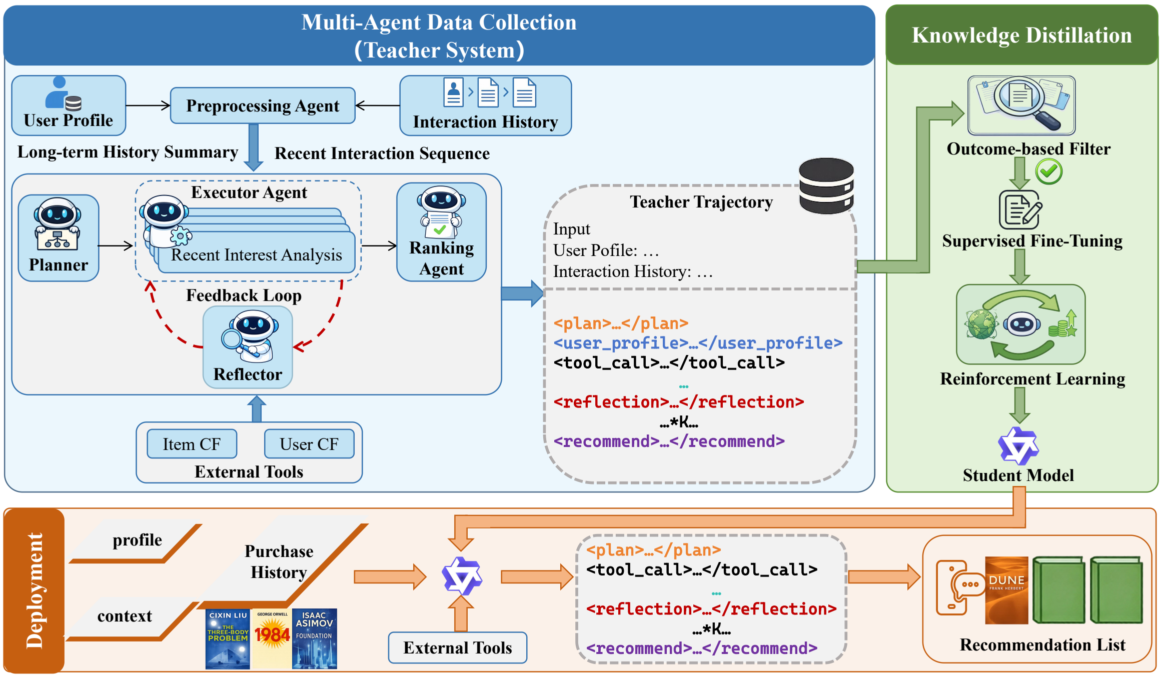


Figure 1: The overall framework of our approach. It consists of two phases: (1) The Multi-Agent Recommender System as a teacher to synthesize reasoning-rich interaction trajectories using external tools; (2) A trajectory-driven internalization pipeline that distills these capabilities into the STAR student model via Supervised Fine-Tuning and reinforcement learning.

3.2 Multi-Agent Teacher System

To derive the reasoning-rich supervision signals required for the formulated task, we first construct the Multi-Agent Recommender System (MARS). This teacher framework functions to demonstrate agentic capabilities and synthesize interaction trajectories that explicitly verbalize latent collaborative signals. However, a critical challenge arises when handling users with extensive interaction histories as processing raw sequences directly often exceeds the inherent context window limits of LLMs.

Input Abstraction Strategy. In response to this constraint, we employ a specialized Long Sequence Abstractor as a data prerequisite. This module utilizes a sliding-window iterative summarization strategy to compress distant historical behaviors while retaining recent interactions in their raw form. The abstraction proceeds recursively where the summary of each historical chunk is fused with the accumulated context from previous chunks. This strategy yields a hybrid input consisting of a global long-term summary followed by a high-fidelity sequence of recent behaviors. For the implementation details, see Appendix E.

Collaborative Reasoning Architecture. Operating on the processed input, the core system follows a Plan-Execute-Reflect paradigm. This architecture begins with the Planner, acting as the strategic brain. Its core task is to analyze the complex user request and decompose the recommendation intent into a sequence of actionable subtasks. These subtasks are then dynamically dispatched to the most appropriate experts in an optimized execution order, ensuring that the reasoning process is tailored to the specific nuances of the user's history.

Specialized Execution Agents. The execution phase is carried out by a group of specialized agents, each focusing on a distinct aspect of user interest. First, the User Profile agent builds a comprehensive profile by extracting user characteristics and potential traits from behavioral records. Second, to handle time-sensitive preferences, separate agents are assigned to Historical and Recent Interest Analysis. They differentiate between stable long-term habits and immediate short-term needs, effectively filtering out noise. Third, the Interest Divergence agent improves recommendation diversity by exploring potential interest directions through semantic expansion. Crucially, these agents actively use the Collaborative Signal Translation mechanism (Section 3.3) to retrieve grounded evidence from the user-item graph.

Reflective Verification and Ranking. To ensure the reliability of the generated content, a Reflector agent is employed to verify the outputs of the execution agents. It checks the results for consistency, rationality, and completeness, ensuring that the reasoning aligns with the user's actual situation. If any issues are detected, the Reflector provides suggestions to guide the revision process. Finally, the Ranking agent executes the last step. It aggregates the comprehensive evidence to rank the candidate items, balancing long-term and short-term needs to produce the optimal recommendation list. For the detailed prompt templates of all agents, please refer to Appendix F.1

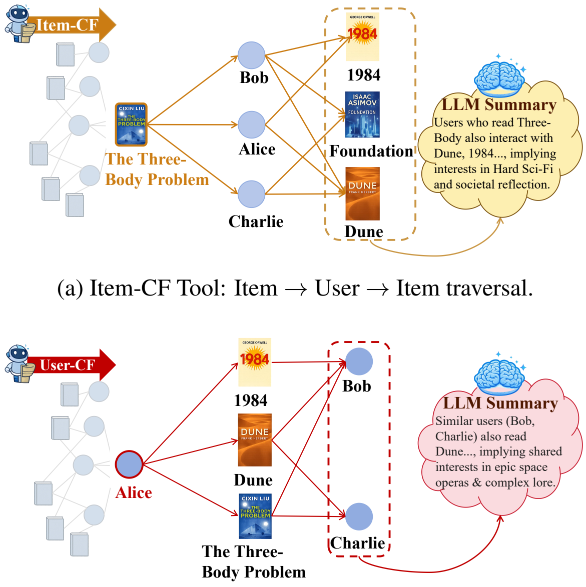
3.3 Collaborative Signal Translation Mechanism

A critical challenge in LLM-based recommendation is bridging the gap between implicit collaborative signals derived from user interactions and the semantic reasoning capabilities of LLMs. To address this, we propose the Collaborative Signal Translation Mechanism. This mechanism serves as a tool interface that retrieves behavioral patterns from the underlying graph structure and translates them into natural language evidence.

Graph Construction as Retrieval Basis. While the sequential view effectively models temporal evolution, it often struggles to capture high-order collaborative relationships. To incorporate these global dependencies, we explicitly construct a global user-item bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ and $\mathcal{E}$ represents the set of observed interactions, where an edge $e_{u,v} \in \mathcal{E}$ exists if user u has interacted with item v. This graph structure is fundamental to our framework, serving as the retrieval basis for the subsequent User-CF and Item-CF tools to identify similar users and co-occurring items via explicit graph traversal.

Item-CF Tool. To reveal latent item associations, this tool traverses the Item $\rightarrow$ User $\rightarrow$ Item path on $\mathcal{G}$ (see Figure 2a). Taking the novel "The Three-Body Problem" as an anchor, the tool identifies users who read it and retrieves other books they frequently purchased, such as "Dune" or "Foundation." This process effectively translates co-occurrence statistics into explicit textual evidence regarding item similarity.

User-CF Tool. Conversely, to leverage user similarity, this tool executes a User $\rightarrow$ Item $\rightarrow$ User traversal on the interaction graph (see Figure 2b). It analyzes items in the user's history to discover other users who have interacted with them. The tool then synthesizes their preferences into a summary, such as "users similar to the target typically prefer hard science fiction with complex world-building." This path explicitly captures user-to-user preference similarities.



(a) Item-CF Tool: Item $\rightarrow$ User $\rightarrow$ Item traversal.

(b) User-CF Tool: User $\rightarrow$ Item $\rightarrow$ User traversal.

Figure 2: Schematic illustration of the graph-based collaborative tools. (a) The Item-CF tool identifies co-occurring items (e.g., retrieving *Dune* via shared readers). (b) The User-CF tool aggregates preferences from similar users to summarize shared interests.

Distinct from approaches that merely return lists of raw IDs or isolated metadata, we employ LLMs to synthesize these structural neighbors into descriptive natural language summaries. Crucially, we can decouple this expensive verbalization from online inference. Since the interaction graph and neighbor sets are fully available prior to serving, we perform this summarization offline and store the resulting natural language evidence as static metadata. Consequently, during online tool invocation, the system merely retrieves these pre-computed contexts, allowing the agent to ground its reasoning in behavioral statistics without incurring the latency overhead of real-time generation.

3.4 Trajectory-Driven Internalization Pipeline

While the multi-agent teacher is powerful, it is computationally expensive for real-time inference. To resolve this, we transfer the teacher's capabilities to STAR via a two-stage internalization pipeline.

Serialized Trajectory Example

<plan>
Decompose user intent into ...
</plan>

<user_profile>
Age: 25, Preference: Hard Sci-Fi, ...
</user_profile>

<historical_analysis>
<tool_call>
Item-CF("Three-Body Problem")
</tool_call>
↪ **<tool_response>**
["Dune", "Foundation"]
</tool_response>
Analysis: User shows strong affinity for space opera and societal reflection...
</historical_analysis>

<reflection>
Tool usage verified. Logic is consistent ...
</reflection>

<recommend>
1. Dune, 2. Foundation, 3. 1984, ...
</recommend>

Figure 3: An example of the serialized chain-of-thought format used for training. Different colors represent distinct agent roles. Crucially, the `<tool_call>` tokens are retained to teach the student model when and how to access collaborative signals.

3.4.1 Trajectory Serialization and Filtering

We serialize the hierarchical communication logs of the teacher into a linear chain-of-thought format y_{chain}. As illustrated in Figure 3, this format explicitly demarcates the reasoning phases using specific tokens, retaining the tool invocation details.

Crucially, retaining the `<tool_call>` tokens ensures that the student model explicitly learns when and how to invoke the collaborative signal translation tools, rather than just memorizing the final answer. To ensure quality, we apply an outcome-based filtering protocol, where only trajectories where the teacher's predicted top-1 item matches the ground truth v_{t+1} are retained for training.

3.4.2 Stage 1: Supervised Fine-Tuning

We first perform Supervised Fine-Tuning (SFT) to initialize STAR using the filtered trajectories. This stage serves as a cold start phase, employing behavioral cloning to teach the student to mimic the planner's task decomposition logic, master the syntax for triggering the collaborative signal translation tools, and follow the structured response format of the teacher.

3.4.3 Stage 2: Policy Optimization via GRPO

While SFT effectively teaches the behavioral format, it is limited to static imitation and lacks the exploration mechanism required to robustly internalize the optimal decision-making logic. To achieve true agentic capability, we further align STAR using Group Relative Policy Optimization (GRPO). The student is treated as a policy π_θ. Specifically, for each input instruction x, we sample a group of G outputs $\{o_1, o_2, \ldots, o_G\}$ from π_θ. The policy is then optimized to maximize a composite reward function $r(o_i, v_{t+1})$ that balances structural strictness and prediction accuracy:

$$r(o_i, v_{t+1}) = r_{fmt}(o_i) + r_{out}(o_i, v_{t+1}), \quad (2)$$

where the total reward consists of two components.

First, the Format Adherence Reward (r_{fmt}) ensures structural integrity. We assign $r_{fmt} = 1$ if the generated trajectory o_i explicitly contains the critical reasoning phases (e.g., `<plan>`, `<reflection>`, `<recommend>`) and maintains valid syntax for tool invocations. Otherwise, a penalty $r_{fmt} = -1$ is applied.

Second, the Tiered Outcome Reward (r_{out}) encourages the model to rank the ground-truth item v_{t+1} as high as possible in the generated ranked list $\hat{y} = [\hat{v}_1, \ldots, \hat{v}_K]$. It is defined as:

$$r_{out}(o_i, v_{t+1}) = \begin{cases} 1, & \text{if } v_{t+1} \in \{\hat{v}_1\} \\ 2/3, & \text{if } v_{t+1} \in \{\hat{v}_1, \hat{v}_2, \hat{v}_3\} \\ 1/3, & \text{if } v_{t+1} \in \{\hat{v}_1, \ldots, \hat{v}_5\} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

By optimizing this objective, STAR learns to refine its reasoning logic to maximize ranking accuracy, effectively internalizing the collaborative reasoning capabilities.

4 Experiments

We evaluate our proposed framework using the standardized protocols from AgentRecBench (Shang et al., 2025) to ensure rigorous and fair comparison.

4.1 Experimental Setup

Datasets We utilize three large-scale real-world datasets provided by the benchmark: Amazon,

Table 1: Performance comparison on Amazon, Goodreads, and Yelp datasets. The table reports the results of our proposed teacher (MARS) and student (STAR) models alongside traditional and agentic baselines. The bottom row indicates the relative performance improvement of STAR over the teacher model.

		Amazon					**Goodreads**					**Yelp**				
		Classic	Cold-Start		Evo-Int		Classic	Cold-Start		Evo-Int		Classic	Cold-Start		Evo-Int	
Category	Method		User	Item	Long	Short		User	Item	Long	Short		User	Item	Long	Short
Traditional	MF	15.0	15.0	15.0	38.4	49.5	15.0	15.0	15.0	29.7	21.3	15.0	15.0	15.0	42.0	65.9
	LightGCN	15.0	15.0	15.0	32.3	59.1	15.0	15.0	15.0	18.0	22.3	15.0	15.0	15.0	31.6	68.9
Agentic RS	BaseAgent	44.0	16.3	14.0	17.3	19.7	20.3	22.3	16.2	32.0	29.7	4.0	4.0	4.0	5.7	4.3
	CoTAgent	39.7	19.3	9.0	17.7	16.0	23.0	20.3	14.5	24.0	16.3	4.3	4.3	4.3	3.3	3.0
	MemoryAgent	43.7	15.3	11.0	18.3	17.7	18.0	16.7	15.5	29.3	29.7	3.7	4.3	4.3	5.0	4.0
	CoTMemAgent	33.3	16.7	12.0	13.3	18.7	17.3	16.7	15.5	23.3	21.3	4.3	3.7	4.3	4.3	4.0
	Baseline666	60.0	50.3	48.7	50.7	71.3	54.7	38.7	49.5	66.0	63.3	7.3	1.3	2.7	0.0	0.0
	DummyAgent	54.0	59.0	45.0	65.0	65.7	56.3	37.7	49.5	66.7	60.7	6.3	1.3	2.7	10.3	6.3
	RecHacker	63.0	59.7	47.0	64.3	68.0	55.0	49.3	46.1	68.7	66.3	7.0	3.0	3.3	9.3	7.7
	Agent4Rec	28.3	45.6	28.0	34.0	46.3	9.3	37.3	11.1	41.3	42.7	7.6	2.7	0.7	10.0	6.0
Ours	MARS	68.4	77.3	68.0	75.7	75.7	63.1	63.7	58.7	74.1	71.3	31.0	9.7	8.1	20.3	13.3
	STAR	**79.0**	**84.0**	**78.0**	**82.3**	**86.0**	**71.7**	**76.3**	**71.4**	**87.3**	**85.3**	**43.0**	**12.3**	**11.3**	**23.0**	**17.7**
	ΔImprov.	+15.5%	+8.7%	+14.7%	+8.7%	+13.6%	+13.6%	+19.8%	+21.6%	+17.8%	+19.6%	+38.7%	+26.8%	+39.5%	+13.3%	+33.1%

Goodreads, and Yelp. These datasets contain millions of interactions, ensuring robust evaluation across diverse domains with varying levels of sparsity. Comprehensive statistics and detailed descriptions are relegated to Appendices A.1 and B.

Evaluation Scenarios We assess model capabilities across three distinct scenarios defined in the benchmark. **Classic Recommendation** tests general performance using complete interaction histories. **Cold-Start Recommendation** evaluates robustness under data sparsity, covering both user-side and item-side cold-start settings. **Evolving-Interest Recommendation** measures adaptability to temporal shifts, examining both long-term preference stability and short-term interest adoption.

Evaluation Metrics. We strictly adhere to the sampled evaluation protocol defined by (Shang et al., 2025). For each test instance, the candidate set comprises the ground-truth item and 19 randomly sampled negative items. We employ Hit Rate at rank k (HR@k) to measure the proportion of test cases where the ground-truth item appears within the top-k positions. To provide a consolidated assessment, we report the average Hit Rate across three cutoffs ($k \in \{1, 3, 5\}$), calculated as $\text{HR}_{avg} = \frac{1}{3} \sum_{k \in \{1,3,5\}} \text{HR@}k$.

Baselines We compare our method against two categories of recommender systems. To ensure fairness, all agentic methods utilize DeepSeek-V3 (DeepSeek-AI, 2024) as the backbone. Detailed settings are in Appendix B.

- **Traditional and Deep Learning:** We include MF (Koren et al., 2009) and LightGCN (He et al., 2020) as representative latent factor and graph-based approaches.
- **Agentic Methods:** This category spans from foundational agents (BaseAgent, CoTAgent, MemoryAgent, CoTMemAgent) to advanced frameworks. Specifically, we include Agent4Rec (Zhang et al., 2024a) and the top-performing solutions from the AgentSociety Challenge (Yan et al., 2025) (Baseline666, RecHackers, DummyAgent).
- **Proposed Variants:** To validate our distillation pipeline, we analyze: 1) MARS: Our multi-agent teacher framework; (2) MARS-Planner: A hybrid baseline that combines the teacher's executors with a planner fine-tuned exclusively on planning trajectories, used to isolate the impact of planning alignment; (3) STAR: The final student model that fully internalizes agentic capabilities.

Implementation Details. Teacher agents utilize DeepSeek-V3 deployed on 16 H20 GPUs, while student models and the planner in MARS-Planner are initialized with Qwen3-8B (Yang et al., 2025). We employ SWIFT (Zhao et al., 2024) for SFT (8 H20; 3 epochs, lr=2e-5, batch size 16) and veRL (Sheng et al., 2024) for GRPO training (16 H20). See Appendix A.2 for full configurations.

4.2 Main Results

Table 1 presents the comparative performance of our proposed framework against all baselines. We analyze the results from two primary perspectives: overall competitiveness and the effectiveness of our distillation strategy across scenarios.

STAR achieves strong performance across all datasets, consistently outperforming both traditional and agentic baselines. Traditional ID-based

methods (MF, LightGCN) perform poorly in Classic and Cold-Start scenarios, with scores remaining around 15.0, underscoring their limitation in processing the rich semantic information required by this benchmark. In contrast, our approach establishes a clear advantage over advanced agentic frameworks. Notably, STAR outperforms the runner-up RecHacker and the challenge winner Baseline666 by a clear margin; for instance, scoring 79.0 in the Amazon Classic scenario compared to RecHacker's 63.0. This suggests that our method synthesizes user preferences more effectively than existing agentic approaches.

Our framework exhibits strong robustness in challenging settings, particularly in Cold-Start and Evolving-Interest scenarios where data sparsity and temporal shifts pose significant difficulties. For example, in the Goodreads Cold-Start User task, STAR achieves a Hit Rate of 76.3, significantly exceeding RecHacker (49.3). Crucially, the student model achieves performance comparable to, and in some cases exceeding, its teacher system (MARS). In the Amazon Classic task, STAR (79.0) outperforms MARS (68.4). This indicates that our distillation pipeline successfully filters noise and transfers high-quality reasoning patterns. As a result, the student generalizes better than the complex multi-agent system itself.

5 Ablation Study

To investigate the contribution of each component in our framework, we conduct ablation studies on the Goodreads dataset. The impact of tool augmentation, training strategies, and teacher components are systematically evaluated. Additionally, the Hybrid baseline (MARS-Planner) is designed to decouple the gains derived from planning alignment versus full capability internalization. Detailed experiments regarding inference scaling are provided in Appendix C.

Superiority of the Student Model As shown in Figure 4 and Table 2, STAR achieves an optimal balance between efficiency and performance. The distinct hierarchy STAR (78.4) > MARS-Planner (69.4) > MARS (66.2) reveals that while partial planner alignment yields moderate gains (+3.2), it still significantly trails the full student (-9.0), confirming that internalizing the complete reasoning pipeline is essential.

Crucially, the student surpasses its teacher—a result we attribute to two factors: outcome-based filtering that removes noisy trajectories, and a unified policy that eliminates error propagation across agent boundaries. Additionally, the performance boost from GRPO (75.5 → 78.4) indicates that reward-driven optimization yields meaningful gains beyond basic behavioral cloning.

Table 2: Component ablation on Goodreads. STAR consistently outperforms both MARS and the Hybrid baseline, validating the effectiveness of the full internalization strategy.

Method	Classic	User-CS	Item-CS	Long	Short	Avg.
			Teacher			
MARS	63.1	**63.7**	58.7	**74.1**	**71.3**	**66.2**
w/o Tools	59.3	56.7	51.5	71.4	69.0	61.6
w/o Planner	**63.5**	60.3	**60.5**	72.1	70.3	65.3
w/o Reflector	60.4	57.3	58.5	68.7	67.3	62.4
			Hybrid			
MARS-Planner	70.0	64.3	59.6	79.0	74.3	69.4
			Student			
STAR	**71.7**	**76.3**	**71.4**	**87.3**	**85.3**	**78.4**
w/o GRPO	70.0	72.7	67.7	85.7	81.3	75.5
w/o Tools	66.0	71.0	60.6	85.3	81.3	72.8
w/o Planner	70.7	73.7	70.7	86.1	83.4	76.9
w/o Reflector	68.7	71.3	62.3	79.7	77.3	71.9

Impact of Data Source Configurations To understand the factors driving this performance, we analyze how removing specific teacher components during data collection impacts the distilled student. We discuss them in descending order of impact:

- **Reflector:** The sharpest performance drop (78.4 → 71.9, $\Delta = 6.5$) occurs when training without reflector traces. Notably, this decline is even more severe than the drop observed in MARS ($\Delta = 3.8$). Although outcome-based filtering ensures the training data still contains correct answers, removing the reflector strips away the self-correction reasoning paths. Consequently, STAR fails to internalize the self-reflection capability, limiting its capacity to recover from intermediate reasoning errors during inference. This confirms that for complex recommendation tasks, supervising the dynamic process of error correction is pivotal for enhancing the accuracy of the final outcome.

- **Tools:** Tool removal causes the second largest decline ($\Delta = 5.6$), suggesting that external retrieval is essential for acquiring explicit collaborative signals (e.g., item attributes). However, the student (72.8) substantially outperforms

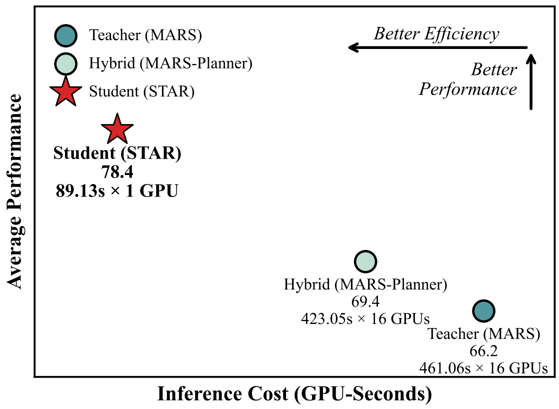


Figure 4: Efficiency-Performance Trade-off. The student model (STAR), indicated by the red star) outperforms the teacher (MARS) while achieving significantly lower inference costs on a single GPU. In contrast, the hybrid MARS-Planner only partially alleviates the resource bottleneck, as it still retains dependency on the high-resource backend.

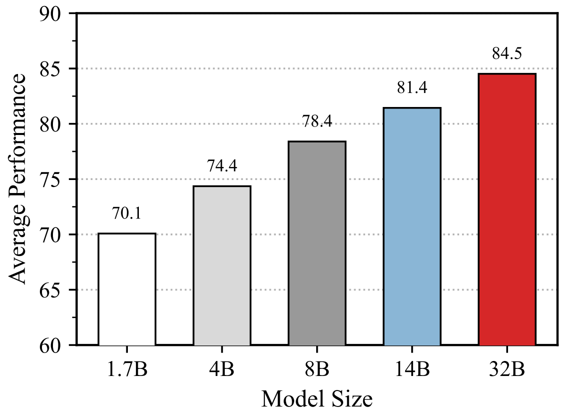


Figure 5: Performance scaling of STAR on the Goodreads dataset across parameter sizes ranging from 1.7B to 32B. The model exhibits consistent performance gains with increasing size.

the tool-less teacher (61.6). This implies that through extensive training on user interaction sequences, our model implicitly learns user-item collaborative patterns. These internalized latent correlations stored in the student's parameters partially offset the information loss caused by the lack of explicit tool retrieval.

- **Planner:** Removing the planner causes a consistent decline (78.4 $\rightarrow$ 76.9, $\Delta = 1.5$), with the largest gap in User Cold-Start ($\Delta = 2.6$). A key insight arises from comparing with MARS: while the planner occasionally introduced noise to the teacher (e.g., lowering performance in Classic tasks from 63.5 w/o Planner to 63.1 Full), STAR consistently benefits from planner-augmented data across all scenarios. This suggests that the distillation process acts as a noise filter: the student successfully captures the beneficial task decomposition logic needed for complex scenarios while bypassing the reasoning redundancy that hampered the teacher in simpler tasks.

These results collectively suggest that self-correction capability and collaborative grounding are the two pillars of successful distillation, while the student effectively refines the planner's logic to achieve universal gains.

Model Scaling Analysis We further investigate the scalability of our distillation framework by training STAR across parameter sizes ranging from 1.7B to 32B. Figure 5 illustrates the performance trend on the Goodreads dataset (comprehensive results for Amazon and Yelp are detailed in Appendix D). We observe a clear scaling law effect where performance improves monotonically with model size. Specifically, the student model gains steadily from 70.1% (1.7B) to 84.5% (32B), demonstrating that the distillation pipeline effectively leverages increased parametric capacity to capture complex collaborative patterns. Extended experiments on Amazon and Yelp confirm that this scaling behavior is robust across diverse domains.

6 Conclusion

In this work, we presented STAR to bridge the gap between semantic reasoning and collaborative signals. Unlike representation enhancers limited by opaque vectors and tool-augmented agents hindered by prohibitive latency, STAR unifies interpretability and efficiency. We achieved this by employing a multi-agent teacher with Collaborative Signal Translation to verbalize latent graph structures, followed by a trajectory-driven distillation pipeline that internalizes this agentic logic into a compact student. Experiments confirm that STAR achieves superior accuracy compared to its teacher while eliminating the coordination overhead of iterative interactions. This study validates that complex reasoning can be effectively internalized for practical recommendation. Future work will focus on further optimizing inference speed for online deployment and extending the framework to diverse recommendation scenarios.

Limitations

Our work has several limitations that warrant future investigation. First, while STAR substantially reduces inference latency compared to the multi-agent teacher (89s vs. 461s), the current speed may still be insufficient for latency-critical applications requiring sub-second responses; further optimization through techniques such as speculative decoding or model quantization remains necessary for online deployment. Second, our offline verbalization strategy requires pre-computing natural language summaries for all users and items, which may pose storage challenges for billion-scale platforms. Third, the evaluation follows the AgentRecBench protocol with 20-item candidate sets; performance on full-corpus ranking scenarios warrants further investigation. Finally, although we demonstrate consistent improvements across three domains (e-commerce, books, and local services), generalization to other verticals such as news or video recommendation remains to be validated.

Regarding the writing process, we utilized AI assistants (e.g., ChatGPT) to assist with grammatical error correction and text polishing. All scientific claims and experimental results were verified by the authors.

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*.

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.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

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.

Yingzhi He, Xiaohao Liu, An Zhang, Yunshan Ma, and Tat-Seng Chua. 2025. Llm2rec: Large language models are powerful embedding models for sequential recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 896–907.

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.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, and 1 others. 2025. How can recommender systems benefit from large language models: A survey. *ACM Transactions on Information Systems*, 43(2):1–47.

Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. 2024. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. *Advances in Neural Information Processing Systems*, 37:26701–26727.

Yuting Liu, Jinghao Zhang, Yizhou Dang, Yuliang Liang, Qiang Liu, Guibing Guo, Jianzhe Zhao, and Xingwei Wang. 2025. Cora: Collaborative information perception by large language model's weights for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12246–12254.

Bo Ma, Hang Li, ZeHua Hu, XiaoFan Gui, LuYao Liu, and Simon Lau. 2025. Agentrec: Next-generation llm-powered multi-agent collaborative recommendation with adaptive intelligence. *arXiv preprint arXiv:2510.01609*.

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*.

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

Yu Shang, Peijie Liu, Yuwei Yan, Zijing Wu, Leheng Sheng, Yuanqing Yu, Chumeng Jiang, An Zhang, Fengli Xu, Yu Wang, and 1 others. 2025. Agentrecbench: Benchmarking llm agent-based personalized recommender systems. *arXiv preprint arXiv:2505.19623*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and 1 others. 2024. Deepseekmath:

Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Jiakai Tang, Yujie Luo, Xunke Xi, Fei Sun, Xueyang Feng, Sunhao Dai, Chao Yi, Dian Chen, Zhujin Gao, Yang Li, and 1 others. 2025. Interactive recommendation agent with active user commands. *arXiv preprint arXiv:2509.21317*.

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.

Junda Wu, Cheng-Chun Chang, Tong Yu, Zhankui He, Jianing Wang, Yupeng Hou, and Julian McAuley. 2024. Coral: collaborative retrieval-augmented large language models improve long-tail recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3391–3401.

Yu Xia, Sungchul Kim, Tong Yu, Ryan A Rossi, and Julian McAuely. 2025. Multi-agent collaborative filtering: Orchestrating users and items for agentic recommendations. *arXiv preprint arXiv:2511.18413*.

Yuwei Yan, Yu Shang, Qingbin Zeng, Yu Li, Keyu Zhao, Zhiheng Zheng, Xuefei Ning, Tianji Wu, Shengen Yan, Yu Wang, and 1 others. 2025. Agentsociety challenge: Designing llm agents for user modeling and recommendation on web platforms. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2963–2967.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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.

Yang Zhang, Keqin Bao, Ming Yan, Wenjie Wang, Fuli Feng, and Xiangnan He. 2024b. Text-like encoding of collaborative information in large language models for recommendation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9181–9191.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2025. Collm: Integrating collaborative embeddings into large language models for recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2024. Swift:a scalable lightweight infrastructure for fine-tuning. *Preprint*, arXiv:2408.05517.

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

Yaochen Zhu, Chao Wan, Harald Steck, Dawen Liang, Yesu Feng, Nathan Kallus, and Jundong Li. 2025b. Collaborative retrieval for large language model-based conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025*, pages 3323–3334.

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

Table 3: Data statistics for Amazon, Goodreads, and Yelp across different training settings

Training Task	Amazon	Goodreads	Yelp
SFT-Full	8646	7958	4673
SFT-w/o Tool	25337	27958	9944
SFT-w/o Planner	8646	7958	4346
SFT-w/o Reflector	8646	7958	4673
SFT-Planner	8646	7958	4673
GRPO	500	470	486

A Training Details and Data Preparation

In this section, we provide comprehensive details regarding the training dataset construction, supervised fine-tuning (SFT) configurations, and reinforcement learning (RL) configurations.

A.1 Dataset Construction

We build our training sample based on the data provided by AgentRecBench (Zhang et al., 2024a), which contains three sub data sources: Amazon, goodreads and yelp.

Training Data Synthesis For each user, we query their user profile from the user table corresponding to the subtask, and query their historical interactions from the specified review table to form a sequence of user historical behaviors. The last item in the behavior sequence sorted by time is taken as the ground truth, and an additional 19 items are taken as negative examples. We perform balanced sampling across multiple task types. To prevent data leakage, we strictly excluded users and their ground truth items that appeared in the test set when constructing the user and their behavior sequences for the training set. The detailed breakdown of training samples across different training settings is presented in Table 3.

SFT Data Construction High-quality SFT samples are distilled using the MARS framework. We perform multiple inference passes for each sample and retain only those samples where the model achieves Hit@1. The final counts of SFT samples are 8,846 for Amazon, 7,958 for Goodreads, and 4,673 for Yelp.

RL Data Construction For reinforcement learning, we sample 8 responses per instance from the SFT-tuned checkpoint and categorize them into three difficulty levels based on the success frequency:

- **Easy**: Success count is between 6 and 7.
- **Medium**: Success count is between 3 and 5.
- **Hard**: Success count is between 1 and 2.

The RL training set is composed according to a 3:4:3 ratio (Easy:Medium:Hard), resulting in 500, 470, and 486 samples for the three datasets, respectively.

A.2 Supervised Fine-Tuning (SFT) and Reinforcement Learning (GRPO) Settings

We employ the Swift (Zhao et al., 2024) framework for full-parameter fine-tuning. The training is conducted on 8 NVIDIA H20 GPUs. Key hyperparameters are summarized in Table 4.

Table 4: Hyperparameters for Supervised Fine-Tuning

Configuration	Value
Model Architecture	Qwen3 8B
Precision	bfloat16
Epochs	2
Global Batch Size	64
Micro Batch Size	2
Global Accumulation	4
Learning Rate	1e-5
Learning Rate Scheduler	Linear with 5% Warmup
Max Sequence Length	16,384
Optimizer	DeepSpeed ZeRO-2

For Group Relative Policy Optimization (GRPO) algorithm, we employ the *VERL* (Sheng et al., 2024) framework. The training is conducted on 16 NVIDIA H20 GPUs. The key parameters for RL training are detailed in Table 5.

Table 5: Hyperparameters for Reinforcement Learning

Hyperparameter	Value
Actor Learning Rate	1e-6
KL Coefficient (β)	0.001
Rollout Temperature	0.7
Group Size (G)	8
Max Prompt Length	6,000
Max Response Length	8,192
Mini-batch Size	4
Tensor Model Parallel (TP) Size	2

B Evaluation Details

B.1 Utilization of Evaluation Datasets

We utilize the official evaluation samples and databases provided by the AgentRecBench chal-

Table 6: Inference-time scaling results on Amazon Dataset.

Method	Score@1	Score@2	Score@4	Score@8	Score@16
Full	81.9	83.4	85.1	86.2	87.7
w/o GRPO	78.2	81.5	83.3	84.7	86.3
w/o Tools	75.1	77.5	78.9	81.0	82.6
w/o Planner	79.9	80.6	83.1	84.7	85.4
w/o Reflector	76.3	78.7	80.7	82.5	85.3

Table 7: Inference-time scaling results on Goodreads Dataset.

Method	Score@1	Score@2	Score@4	Score@8	Score@16
Full	78.4	79.2	80.4	81.9	83.5
w/o GRPO	75.5	77.6	79.1	81.0	82.7
w/o Tools	72.8	73.2	74.9	76.4	78.4
w/o Planner	76.9	78.5	79.7	81.5	83.0
w/o Reflector	71.9	73.1	74.8	77.7	80.5

Table 8: Inference-time scaling results on Yelp Dataset.

Method	Score@1	Score@2	Score@4	Score@8	Score@16
Full	21.5	22.3	22.7	23.1	23.8
w/o GRPO	20.1	21.0	21.7	22.7	23.0
w/o Tools	18.9	19.3	19.7	20.1	22.1
w/o Planner	20.9	21.3	21.7	22.1	22.7
w/o Reflector	17.5	17.7	18.2	19.6	21.8

lenge. This benchmark structures specialized database tables tailored to various sub-tasks. Each raw evaluation sample comprises a “user_id”, a candidate list, and the ground truth. Our primary objective is to determine the rank of the ground truth within the model-generated recommendation list to calculate the Hit-Ratio.

Specifically, starting from the given “user_id”, we retrieve user profile features and historical interaction records from the designated user and review tables, respectively, while fetching item metadata from the item table. To ensure the integrity of the evaluation and prevent data leakage, we strictly adhere to the benchmark’s protocols: entries related to the ground truth are filtered out from the review table. Furthermore, to avoid look-ahead bias, we truncate user behavioral sequences after their first interaction with the ground truth. We evaluate all sub-tasks across every data source, utilizing all 100 available samples per sub-task, amounting to a total of 1,500 evaluation samples.

B.2 Model Deployment and Evaluation

For the teacher model, we employ DeepSeek-V3 (DeepSeek-AI, 2024), deployed on a cluster of 16 NVIDIA H20 GPUs. We utilize a tensor parallelism degree of 16 with inference performed in BF16 precision. For the student models, we utilize the Qwen3 series (Yang et al., 2025) (ranging from 1.7B to 32B parameters), each deployed on a single NVIDIA H20 GPU using BF16 precision.

Model inference is executed locally to ensure precise latency measurement. We strictly adhere to the official chat templates for each respective open-source model. Regarding generation hyperparameters, we set the $temperature$ to 0.7 and top_p to 0.95.

C Inference-Time Scaling via Best-of-k Evaluation

We investigate the inference-time scaling properties of our model using a Best-of-k strategy. As detailed in Tables 6–8, performance demonstrates a consistent monotonic improvement as the sample budget k increases from 1 to 16. This validates the efficacy of *test-time compute* in the recommendation domain: by allotting more computational budget during inference, the model can explore a broader spectrum of reasoning trajectories.

This scaling effect stems from the diversity of the generated rationales. While a greedy decoding strategy (equivalent to $k = 1$) might commit to a suboptimal reasoning path early on due to local probability spikes, sampling k times allows the model to correct potential hallucinations or logical pitfalls by covering diverse tool-use patterns.

Furthermore, the ablation comparison highlights the critical role of our training pipeline. The Full model not only starts with a higher baseline at $k = 1$ but also maintains a robust growth rate compared to the w/o GRPO variant. This indicates that while Supervised Fine-Tuning (SFT) teaches the model how to use tools, the Group Relative Policy Optimization (GRPO) aligns the probability mass towards more effective and accurate reasoning chains, making the sampling process more efficient. Conversely, the w/o Tools variant shows significantly lower performance ceilings, confirming that reasoning without grounded evidence limits the potential gains from scaling inference compute.

In practice, this property offers a flexible trade-off between latency and accuracy. For scenarios requiring real-time response, a smaller k yields competitive performance; for high-stakes recommendations where precision is paramount, increasing k unlocks the model’s potential without retraining.

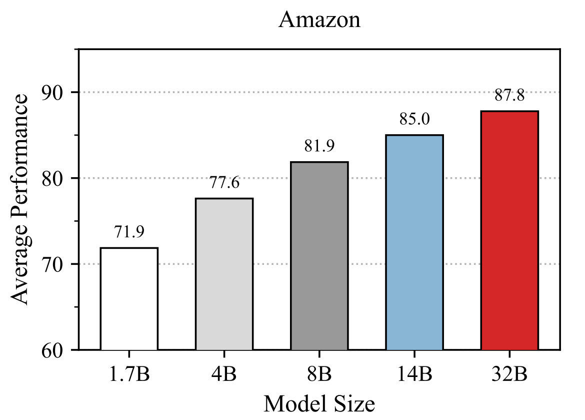


Figure 6: Performance scaling of STAR on the Amazon dataset across parameter sizes ranging from 1.7B to 32B. The model exhibits consistent performance gains with increasing size.

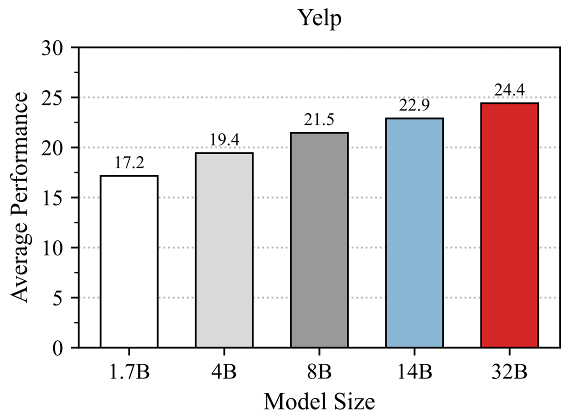


Figure 7: Performance scaling of STAR on the Yelp dataset across parameter sizes ranging from 1.7B to 32B. The model exhibits consistent performance gains with increasing size.

D Model Size Scaling of STAR across Different Data Source

To verify the universality of our trajectory-driven internalization pipeline, we investigate the scalability of STAR on two additional heterogeneous domains: Amazon and Yelp.

As illustrated in Figure 6 and Figure 7, we observe a consistent monotonic improvement in recommendation performance as the student model size scales from 1.7B to 32B parameters.

This trend corroborates the core insight: the efficacy of trajectory-driven distillation relies heavily on the student's reasoning capacity. As parameter size increases, the student becomes increasingly proficient at internalizing the complex Plan-Execute-Reflect logic, transforming from a mere pattern matcher into a reasoning-capable recommender. This cross-domain consistency further validates that our framework is a generalizable solution for empowering efficient models with deep reasoning capabilities.

E Implementation Details of Preprocessing Agent

When utilizing Large Language Models (LLMs) for recommendation tasks, the input length limitation of LLMs poses a critical challenge in processing lengthy user behavior sequences, as the complete sequence cannot be fed into the model at once. To address this issue, a preprocessing agent for compressing user behavior sequences is proposed, and the specific process is illustrated in Figure 8. First, the user's historical behaviors are sorted in chronological order. Subsequently, the sorted behavior sequence is partitioned into consecutive groups, where each group contains m sequential behaviors. The compression of these grouped behaviors is achieved through an iterative summarization mechanism based on the LLM, and the specific process is as follows:

1. The LLM is first instructed to generate a summary for the first group of behaviors;

2. On the basis of the previously generated summary summary, the LLM is then instructed to integrate the information of the second group of behaviors and generate a new summary;

3. This iterative summarization process is continued: for each subsequent group of behaviors, the LLM generates a new summary by fusing the summary obtained from the previous iteration and the current group of behaviors;

4. The iterative summarization is terminated when all behavior groups except the last one have been processed.

It should be emphasized that the last group of behaviors is not involved in the summarization process. For the subsequent recommendation task, the final summary generated from the above iterative process and the last group of behaviors are concatenated as a whole input, which is then fed into the downstream MARS system to complete the recommendation task.

Prompt for Preprocess Agent

1. Role & Mission
You are a long behavioral sequence summary agent in a multi-agent system designed to perform recom-

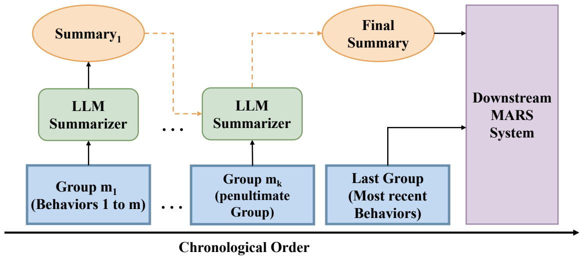


Figure 8: Running process of the preprocessing agent

mendation tasks on the goodreads dataset, your task is to summarize the user's core behaviors based on the provided time-segmented behavioral data.
2. Detailed Description
1. Focus on the provided behavioral data and fully extract core behavioral information: Prioritize the user's key behavior types, the chronological logic between behaviors, and important behavioral nodes that outline the action trajectory. Eliminate irrelevant and redundant content, retaining only core elements.2. Conduct in-depth analysis of the time-segmented behavioral data to capture comprehensive details: Perform detailed parsing of the provided time-segmented behavioral data, focusing on identifying three types of content—all behavioral types presented, behavioral frequency distribution characteristics (such as concentrated or scattered interactions), and iconic key action events.3. Screen high-value behavioral details: Independently assess the value of information from the provided data, filter duplicate descriptions (such as repeated records of similar trivial behaviors), and retain details most meaningful for restoring the complete behavioral context.4. Construct a chronological summary to ensure logical coherence: The generated summary must follow a time-based structure, clearly presenting the user's behavioral sequence and internal connections to avoid disjointed descriptions. Keep the summary within 500 words to ensure conciseness, clarity, and focus. Only objectively present behavioral facts and their chronological connections—do not involve subjective inferences about the user's preferences, needs, or intentions.5. Associate item identifiers to enrich behavioral dimensions: Items interacted with in user behaviors are all accompanied by item_ids. Appropriately reference these identifiers in the summary to make behavioral descriptions more specific.
3. User input Specifications
1. User Information
2. Time-segmented behavioral data
4. Output Requirements
Strictly following a two-phase process: first, internal reflection and analysis (Thinking), and second, structured output
1. Thinking: You are free to organize your logic, analyze the problem, and derive reasoning paths
2. Final Answer: The generated, complete summary content of the user's behavioral sequence
- Format constraints: The summary content is enclosed within the tags <SUMMARY></SUMMARY>
For example:
« Your Thinking Process »
<SUMMARY> « Your Summary » </SUMMARY>

F Prompt templates and Examples

F.1 System prompt for each agent component in the MARS

Prompt for Planner

1. Role & Mission
You are a planner agent in a multi-agent system designed to perform recommendation tasks on the goodreads dataset, your task is to decompose a given recommendation task into multiple subtasks and assign those subtasks to different agents.
2. Detailed Description
The agents you can choose from include:
1. User_Profile_Summary: Integrate users' basic information, behavioral data and preference feedback, extract core feature tags, and build a comprehensive user feature model
2. Historical_Interest_Analysis: Analyze users' long-term behavioral data, filter accidental noise, and identify stable core interest directions
3. Recent_Interest_Analysis: Focus on users' short-term behavioral dynamics, capture preference fluctuations and new trends, and inject fresh directions into recommendations
4. Interest_Divergence_Reasoning: Expand semantically based on known interests, explore users' potential related interest directions, and enrich recommendation diversity
Please select the agents you consider necessary and specify the order in which they should be executed.

Prompt for User Profile Summary

1. Role & Mission
You are a user profile summary agent in a multi-agent system designed to perform recommendation tasks on the goodreads dataset, your task is to extract and generate a comprehensive and precise set of user characteristics based on multi-dimensional user data (including basic information, behavior records, and reviews). This is to construct a user profile, which will serve as the core basis for recommendation task.
2. Detailed Description
Infer potential user persona dimensions (including age, gender orientation, geographical location, professional field, educational level, purchasing power, and consumption philosophy), core personality traits (such as decision-making style, interaction preference, and emotional tendency), and explicit and latent interest preferences (including long-term stable interests, recent dynamic interests, and unarticulated potential needs), based on the user's provided basic information, interaction history reviews.Adhere to the principle of "outputting only what can be analyzed, and not forcibly supplementing what cannot be inferred." Dimensions lacking effective data support or that cannot be reasonably inferred should not be included in the final answer, thus ensuring the objectivity and credibility of the analysis results.

Prompt for Historical Interest Analysis

1. Role & Mission
You are a Historical interest analysis agent in a multi-agent system designed to perform recommendation tasks on the goodreads dataset, your task is to identify stable and persistent areas of user interest to generate a set of long-term interest tags. This serves as the core basis for the recommendation system, reflecting the user's essential needs, and helps prevent recommendation bias caused by interference from short-term behaviors.
2. Detailed Description
Define the time scope for long-term behavior and filter the user's full behavioral data within this period. Exclude anomalous and low-value data. Through in-depth behavioral analysis and interest correlation mining, summarize the long-term and stable interest areas.

Prompt for Recent Interest Analysis

1. Role & Mission
You are a Recent interest analysis agent in a multi-agent system designed to perform recommendation tasks on the goodreads dataset, your task is to precisely identify the latest changes, emerging needs, and temporary focuses in user interest preferences, generating a set of recent interest tags with strong timeliness. This supplements the recommendation system with the dimension of immediate needs.
2. Detailed Description
Define the recent data time scope and focus on the user's instantaneous behavioral data and immediate feedback within this period. Extract the user's short-term core characteristics through behavioral immediacy analysis and interest novelty assessment.

Prompt for Interest Divergence Reasoning

1. Role & Mission
You are a Recent interest analysis agent in a multi-agent system designed to perform recommendation tasks on the goodreads dataset, your task is to expand semantically based on known interests, explore users' potential related interest directions, and enrich recommendation diversity.
2. Detailed Description
Perform semantic parsing on the user profile and known interests, and achieve interest expansion through a multi-dimensional inference model. Focus on three major inference directions: Association Extension, Scenario Derivation, and Group Commonalities, ensuring the relevance and rationality of the potential interests. This provides diverse material for recommendation ranking.

Prompt for Reflection

1. Role & Mission
You are a Reflector agent in a multi-agent system designed to perform recommendation tasks on the goodreads dataset, your task is to reflect on and verify the execution output of the preceding agents, identify the problematic nodes and provide the suggestions.
2. Detailed Description
Receive the output results from all preceding agents, and conduct systematic inspection and reflective verification on the output content of each agent by combining the original user information and comment data. The core is to construct a verification system around three key dimensions: consistency, rationality, and completeness:
1. Consistency: Focus on verifying the logical self-consistency between preceding results to avoid data conflicts; simultaneously validate the internal consistency of individual agent results—for example, the confidence level of interest tags must accurately match the underlying behavioral supporting evidence.
2. Rationality: Based on actual user scenarios and general common-sense logic, conduct in-depth judgment on the actual fit between each output result and user needs, ensuring that interest mining and profile portrayal align with the user's true situation.
3. Completeness: Ensure that the preceding outputs fully cover the core information dimensions required for the recommendation task without missing key data, and can provide sufficient and reliable decision support for the subsequent candidate sample ranking process.
If the inspection is not passed, identify the issues and provide the suggestions.

Prompt for recommendation Ranking

1. Role & Mission
You are a ranking agent that performs the final step of the recommendation task, which is the ranking task on the goodreads dataset, your task is to use comprehensive information and rigorous reasoning to accurately rank 20 candidate items from highest to lowest, providing users with the most suitable results.
2. Detailed Description
Please conduct priority evaluation on the 20 designated candidate items based on the provided user information, user behavior data, and analysis conclusions from other agents. During the evaluation process, you need to focus on balancing the core attributes of the user profile, the stable preferences of long-term interests, the immediate demands of short-term interests, and the potential needs of interest extension. Strictly in accordance with the balancing results, output a unique sorting result in descending order — this result must accurately match the degree of alignment between the items and the user's diverse needs.

F.2 Prompt and output of Item-CF and User-CF

Item CF Example

System Prompt
1. Role & Mission
You are a a collaborative filtering result summarization agent to perform recommendation tasks on the goodreads dataset, your task is to perform in-depth analysis, summarization, and classification of a set of collaboratively related items (denoted as the "collaborative items") based on a given target item and a set of collaboratively related items selected through collaborative filtering.
2. Detailed Description
1. Analyze the shared characteristics of the collaborative items: Perform a comprehensive analysis of all related items, considering (but not limited to) semantic similarity, functional similarity, usage scenarios, content type, and theme
2. Cluster and generalize based on common features: Group the related items into several categories according to their shared characteristics. The number of categories is not fixed; determine it based on the diversity and strength of commonalities among the items. Each category must include a clear semantic explanation describing what unites the items in that group

3. Generate concise category names: Each category should have a short, accurate name that reflects its core characteristics
3. User input Specifications
1. Target item
2. Collaborative items: A set of collaboratively related items selected through collaborative filtering
4. Output Requirements
Strictly following a two-phase process: first, internal reflection and analysis (Thinking), and second, structured output
1. Thinking: You are free to organize your logic, analyze the problem, and derive reasoning paths
2. Final Answer: You must present your conclusion enclosed within <Answer></Answer> tags to ensure it can be parsed directly.
For example:
« Your Thinking Process »
<Answer>
Category 1: <Category Name A>
Category Summary: A short explanation describing the shared characteristics of this group of items.
Representative Items: item_id of up to 5 representative items
...
</Answer>

User Prompt
> Item Key Definition:
> item_id (Unique book identifier)
> title (Book title with series info)
> authors (Author(s) of the book)
> average_rating (Average user rating (0-5))
> ratings_count (Number of user ratings)
> text_reviews_count (Number of text reviews)
Target Item
item_id: 739484 title: That's Not My Dinosaur (Usborne Touchy-Feely Books) authors: ['author_id': '34407', 'role': ''] average_rating: 4.08 ratings_count: 1290 text_reviews_count: 68 # Collaborative items
item_id: 1187586 title: Squib authors: ['author_id': '157819', 'role': ''] average_rating: 3.50 ratings_count: 48 text_reviews_count: 6
item_id: 885497 title: The Tale of Squirrel Nutkin authors: ['author_id': '11593', 'role': ''] average_rating: 4.03 ratings_count: 5314 text_reviews_count: 216

Model Output
Looking at the collaborative items in relation to the target item "That's Not My Dinosaur," I notice several patterns emerge. The target item is a tactile board book for very young children featuring dinosaurs. The collaborative items are quite diverse, but I can identify several groups based on shared characteristics like target audience, themes, and format. **Category 1: Animal-Themed Picture Books** This is the largest and most direct connection to the target book. These are picture books where animals are central characters, often with simple, heartwarming stories aimed at young children. Like the target book, they use animals to engage children. - *The Pigeon Wants a Puppy!* - *A Sick Day for Amos McGee* - *The Tale of Squirrel Nutkin* - *Love You Forever* (features a mother's love, often illustrated with animal motifs) - *Puppies, Pussycats & Other Friends* **Category 2: Interactive & Activity-Focused Books** These books go beyond a simple narrative and encourage direct interaction, creativity, or problem-solving. This connects to the "Touchy-Feely" interactive element of the target book. - *Adventures in Cartooning: How to Turn Your Doodles Into Comics* - *The Unbreakable Code (Book Scavenger, #2)* (involves puzzles) - *Babymouse: Dragonslayer* (engages with comic/graphic novel format) - *The Case of the Good Boy (Bad Machinery #2)* (mystery-solving) **Category 3: Whimsical & Imaginative Stories** This group consists of stories with a strong element of fantasy, gentle magic, or charming, unusual premises. They share the playful and imaginative spirit found in a book about a dinosaur with textured patches. - *The Chocolate Touch* - *Momo* - *Greenglass House* - *Something Fishy at MacDonald Hall* (lighthearted school caper) **Category 4: Seasonal & Holiday-Themed Books** These books are centered around specific holidays or seasons, providing a thematic connection that parents might seek out, similar to how a dinosaur book might be part of a thematic collection. - *Angelina's Christmas* - *One Halloween Night* **Category 5: Emotionally Resonant & Poetic Books** This smaller group focuses on conveying deep emotions, tackling meaningful themes, or using a poetic style. They share the goal of connecting with a child on an emotional level, albeit for a slightly older audience than the target book. - *To This Day: For the Bullied and Beautiful* - *Through Endangered Eyes: A Poetic Journey into the Wild* - *Rifka Takes a Bow* (theater/immigrant experience)
<Answer>
Category 1: Animal-Themed Picture Books Category Summary: Picture books featuring animals as central characters, often with simple, engaging narratives aimed at young children, directly connecting to the dinosaur theme of the target item. Representative Items: 1908511, 7268995, 885497, 310259, 7694275 Category 2: Interactive & Activity-Focused Books Category Summary: Books that encourage direct reader interaction, creativity, or problem-solving, aligning with the tactile, interactive "Touchy-Feely" nature of the target book. Representative Items: 5852045, 25944803, 7200097, 18710631 Category 3: Whimsical & Imaginative Stories Category Summary: Stories characterized by fantasy, gentle magic, or charmingly unusual premises, sharing the playful and imaginative spirit suitable for children. Representative Items: 88455, 68811, 18222716, 37914 Category 4: Seasonal & Holiday-Themed Books Category Summary: Books centered around specific holidays or seasons, appealing to readers looking for thematic collections similar to children's book series.

983

Representative Items: 924104, 1342645 Category 5: Emotionally Resonant & Poetic Books Category Summary: Books that focus on conveying deep emotions, meaningful themes, or using a poetic style to connect with readers on a personal level. Representative Items: 21561020, 6084673, 17364849
</Answer>

User CF Example

System Prompt
1. Role & Mission
You are a a collaborative filtering result summarization agent to perform recommendation tasks on the goodreads dataset, your task is to perform in-depth analysis, summarization, and classification of a set of preference-related items (denoted as the "preference items") based on a set of items liked by users with similar behavioral preferences to the target user (selected through user-based collaborative filtering).
2. Detailed Description
1. Analyze the shared characteristics of the preference items: Perform a comprehensive analysis of all these items, considering (but not limited to) semantic similarity, functional similarity, usage scenarios, content type, theme, and the correlation with the target user's historical preferences
2. Cluster and generalize based on common features: Group the preference items into several categories according to their shared characteristics. The number of categories is not fixed; determine it based on the diversity and strength of commonalities among the items. Each category must include a clear semantic explanation describing the core feature that unites the items in that group, and briefly clarify the connection between this feature and the target user's potential preferences
3. Generate concise category names: Each category should have a short, accurate name that reflects its core characteristics and is easy to understand
3. User input Specifications
Preference items: A set of items liked (or with positive interactions) by the similar-behavior users
4. Output Requirements
Strictly following a two-phase process: first, internal reflection and analysis (Thinking), and second, structured output
1. Thinking: You are free to organize your logic - first clarify the core preference characteristics of the target user based on their historical behaviors, then analyze the connection between the similar-behavior users and the target user, further explore the common features of the preference items, and finally complete the clustering and reasoning of the items
2. Final Answer: You must present your conclusion enclosed within <Answer></Answer> tags to ensure it can be parsed directly. The content should include category name, category summary, preference relevance explanation, and representative items for each category
For example:
« Your Thinking Process »
<Answer>
Category 1: <Category Name A>
Category Summary: A short explanation describing the shared characteristics of this group of items.
Representative Items: item_id of up to 5 representative items
...
</Answer>

User Prompt
> Item Key Definition:
> item_id (Unique book identifier)
> title (Book title with series info)
> authors (Author(s) of the book)
> average_rating (Average user rating (0-5))
> ratings_count (Number of user ratings)
> text_reviews_count (Number of text reviews)
Collaborative items
item_id: 3674269 title: Perahu Kertas authors: [’author_id’: ’167915’, ’role’: ”] average_rating: 4.18 ratings_count: 326 text_reviews_count: 16
item_id: 22551791 title: When You Were Born authors: [’author_id’: ’24450’, ’role’: ”] average_rating: 3.95 ratings_count: 45 text_reviews_count: 9
item_id: 10455693 title: Amelia Rules! Volume 7: The Meaning of Life... and Other Stuff (Amelia Rules! #7) authors: [’author_id’: ’266508’, ’role’: ”] average_rating: 4.27 ratings_count: 333 text_reviews_count: 64
...

Model Output
Let me analyze these preference items to understand the target user's potential preferences through the lens of similar-behavior users. First, looking at the items: - "Perahu Kertas" - Indonesian novel, coming-of-age story - "When You Were Born" - likely children's book about birth/family - "Amelia Rules! Volume 7" - children's graphic novel series - "The Wonder Book" - children's poetry/illustrated book - "Um Urso Chamado Paddington" - Portuguese edition of

Paddington Bear classic - "Cookies: Bite-Size Life Lessons" - children's book with moral lessons - "The Count of Monte Cristo" - classic literature (adapted version) - "Brave Potatoes" - children's picture book - "Tailypo: A Newfangled Tall Tale" - children's folklore/tall tale - "A Child's Garden of Verses" - classic children's poetry - "Lewis Carroll's Alice in Wonderland" - classic children's literature - "The Great Googly Moogly" - children's picture book - "Emergency!" - likely children's book about emergencies - "Mittens" - children's book (likely about kittens) - "Black Cat" - children's picture book - "Bear" - children's book about bears - "The Power of the Fortrex" - LEGO NEXO Knights children's reader - "Cock-A-Doodle Quack! Quack!" - children's animal book - "Sensual Math: Poems" - poetry collection (adult/young adult) - "The Girl in the Tower" - children's fantasy/adventure Looking at the patterns: 1. Strong presence of children's literature (18 out of 20 items) 2. Mix of picture books, chapter books, and poetry 3. International/cross-cultural elements (Indonesian, Portuguese editions) 4. Classic children's literature alongside contemporary works 5. Educational/moral themes in many titles 6. Animal characters are very common I can identify 4 main categories based on the common characteristics: 1. Children's Picture Books with Animal Characters 2. Classic Children's Literature 3. Children's Educational & Moral Stories 4. Poetry Collections for Young Readers The target user likely has preferences for children's literature, enjoys books with educational value, appreciates animal characters, and has interest in both contemporary and classic works. The similar-behavior users suggest the target user may be a parent, educator, or someone who enjoys quality children's content.
<Answer>
Category 1: Children's Picture Books with Animal Characters Category Summary: Colorful picture books featuring animal protagonists and simple narratives designed for young readers Preference Relevance: Appeals to users who enjoy gentle, visually engaging stories with relatable animal characters and themes suitable for children Representative Items: 25180735, 1070755, 2406681, 4793875, 671009 Category 2: Classic Children's Literature Category Summary: Timeless children's stories and adaptations of literary classics that have enduring appeal across generations Preference Relevance: Indicates interest in foundational children's literature with proven educational and entertainment value Representative Items: 7527980, 118458, 8390109, 25180735, 652455 Category 3: Children's Educational & Moral Stories Category Summary: Books that combine storytelling with life lessons, character development, and educational content for young readers Preference Relevance: Suggests preference for content that provides both entertainment and developmental value for children Representative Items: 821582, 1385734, 10455693, 28894946, 7055794 Category 4: Poetry Collections for Young Readers Category Summary: Verse-based books and poetry collections specifically curated or adapted for children and young adult audiences Preference Relevance: Reflects appreciation for lyrical language, rhythm, and creative expression in children's literature Representative Items: 118458, 1149160, 7055794, 3674269, 29102827
</Answer> 986

F.3 An example of the input and corresponding output of the STAR system 987

Serialized Trajectory Example

System Prompt
Integrated Recommendation Agent
1. Role & Mission
You are an autonomous agent responsible for the end-to-end recommendation workflow. Your goal is to rank candidate items for a user by executing a structured multi-agent process: **Plan** → **Execute Sub-tasks** → **Reflect & Verify** → **Error Correction (if needed)** → **Final Ranking**.
2. Global Protocol
1. **Sequence**: Strictly follow the phase order. Do not skip phases.
2. **Format**: For EVERY step, you must output in this exact format:
`<think>` [Your reasoning logic here] `</think>`
`<JSON>` [Structured output data here] `</JSON>`
3. **No Chatter**: Do not output any text outside the tags.
3. Workflow Specification
Phase 1: Planning
Analyze user inputs and candidate items. Determine the execution order of the following sub-tasks:
1. `User_Profile_Summary`: Build user persona.
2. `Historical_Interest_Analysis`: Analyze long-term stable interests.
3. `Recent_Interest_Analysis`: Capture short-term trends.
4. `Interest_Divergence_Reasoning`: Explore potential interest expansions.
Output: A list of selected sub-task names in order.
Phase 2: Sub-task Execution
Execute the sub-tasks planned in Phase 1. Use external tools (UserCF/ItemCF) if necessary.
- **Goal**: Extract comprehensive insights (demographics, preferences, intent) based on data.
- **Output**: A list of strings containing key analysis conclusions.
Phase 3: Reflection
Verify the Phase 2 outputs for **Consistency**, **Rationality**, and **Completeness**.
- **Pass**: Output `{"correct": "yes"}`.
- **Fail**: Output `{"correct": "no", "problematic_agent": [{"agent_name": "...", "suggestion": "..."}]}`.
Phase 4: Error Correction (Conditional)
If Phase 3 failed, re-execute the problematic sub-tasks following the suggestions.
Output: Corrected analysis results.
Phase 5: Recommendation Ranking 988

Evaluate the 20 candidate items against the analyzed user profile and interests.
Output: A list of `item_id` strings sorted by priority (highest to lowest).
4. Input Data
(User Information, Interaction History, and Candidate Items will be provided below) """
Phase 1: Task Planning
« Thinking » <JSON>["Sub-task 1", "Sub-task 2", ...]</JSON>
Phase 2: Sub-task Execution
Sub-task 1: [Sub-task Name]
« Thinking » <JSON>[Sub-task 1 Output Result]</JSON>
Sub-task 2: [Sub-task Name]
« Thinking » <JSON>[Sub-task 2 Output Result]</JSON> ...
Phase 3: Reflection
« Thinking » <JSON>[Reflection Result]</JSON>
Phase 4: Error Correction
Re-executed Sub-task: [Problematic Sub-task Name 1]
« Thinking » <JSON>[Corrected Sub-task 1 Output Result]</JSON>
Re-executed Sub-task: [Problematic Sub-task Name 2]
« Thinking » <JSON>[Corrected Sub-task 2 Output Result]</JSON> ...
Phase 5: Recommendation Ranking
« Thinking » <JSON>[Candidate Item Ranking Result]</JSON>
"""

User Prompt
User Information
> User Key Definitions:> user_id: User's unique identifier. (Value: `723a2a35783fadad2a4f2e00c5e639d7`)

Long-term User Behavior Summary (2012-06-08 to 2015-02-12)
Following the review of "Toes, Ears, & Nose!" (item_id: 155770) on January 28, 2015, the user continued without interruption.
[... Intervening summary content ...]
The sequence concluded on February 12 with "Clifford The Small Red Puppy" (item_id: 615244), an emotional origin story of the character.

User Behavior (Recent Activity: 2015-02-12 to 2016-03-24)
> Item Key Definition:
> item_id (Unique book identifier)
> title (Book title with series info)
> authors (Author(s) of the book)
> average_rating (Average user rating 0-5)
> ratings_count (Number of user ratings)
> text_reviews_count (Number of text reviews)
1. Review time: 2015-02-12-06PM item_id: 14473 title: Merry Christmas, Amelia Bedelia ... user review text: "I remember reading this book while a was just a wee child myself..."
[... Intervening history items ...]
15. Review time: 2015-02-24-09AM item_id: 3397729 title: The Sorcerer's Apprentice ... user review text: "Basically, this book is a retelling of the famous Disney's Fantasia..."

Candidate Item Information
1. item_id: 31767393 title: Han Solo (2016) #3 (of 5) authors: Marjorie Liu et al. ...
[... Intervening candidate items ...]
20. item_id: 22017603 title: Like A Box Of Chocolates authors: D.J. Mckay ...

Model output

<plan>
<Thinking>Analyzing the user's history (2012-2016), they show a consistent preference for educational children's books, moral lessons, and series continuity (e.g., "Little Critter"). The execution sequence is:
1. User_Profile_Summary: Build a comprehensive user preference model.
2. Historical_Interest_Analysis: Identify stable long-term interests.
3. Recent_Interest_Analysis: Capture short-term trends and specific themes.
4. Interest_Divergence_Reasoning: Explore related themes for diversity.
</Thinking> <JSON>["User_Profile_Summary", "Historical_Interest_Analysis", "Recent_Interest_Analysis", "Interest_Divergence_Reasoning"]</JSON>
</plan>

<user_profile>
<Thinking> I will analyze the user's profile based on their reading history and reviews. Then, I will use collaborative filtering to understand their preference categories. </Thinking>

989

```
<tool_call>
{"name": "UserCF", "arguments": "{"user_id": "723a2a35783fadad2a4f2e00c5e639d7"}"}
</tool_call>
↪ <tool_response>
{"result": "Category 1: Children's Picture Books (Simple stories for family reading); Category 2: Fantasy Adventure Series (Immersive world-building); Category 3: Classic Literature (Foundational works)."}
</tool_response>
Based on the tool results and reviews, the user is a parent who prioritizes educational content, moral lessons, and bright illustrations for their child.
<JSON>["Parent with young child", "Values educational content", "Prefers bright illustrations", "Appreciates character development"]</JSON>
</user_profile>

<historical_analysis>
<Thinking> Analyzing the 2012-2015 timeline reveals stable interests in educational children's literature. The user consistently prioritizes moral development (responsibility, coping) and practical life skills. Key patterns include strong engagement with character-driven series like "Little Critter" and "Berenstain Bears," along with a distinct preference for seasonal themes and interactive formats. These interests form a persistent baseline for recommendations. </Thinking>
<JSON>["Educational Children's Literature", "Moral Development", "Seasonal Books", "Interactive Books", "Character-Driven Series", "Classic Retellings"]</JSON>
</historical_analysis>

<recent_analysis>
<Thinking> Analyzing behavior from 2015 to 2016 reveals a distinct shift towards specific series and seasonal themes. The user demonstrates intense engagement with the "Little Critter" series, reading ten books in rapid succession. New specific interests have emerged around Fall festivals and Earth Day environmental themes, while the foundational preference for interactive books and moral life skills remains strong. </Thinking>
<JSON>["Little Critter Series Engagement", "Fall Seasonal Literature", "Environmental Education", "Interactive Tactile Books", "Character Continuity"]</JSON>
</recent_analysis>

<interest_divergence>
<Thinking> I will explore semantic expansions based on UserCF and ItemCF patterns to identify potential new interests. Let me start by examining the user's reading patterns and then use collaborative filtering to find related interests. </Thinking>
<tool_call>
{"name": "UserCF", "arguments": "{"user_id": "723a2a35783fadad2a4f2e00c5e639d7"}"}
</tool_call>
↪ <tool_response>
{"result": "Category 1: Children's Picture Books (Heartwarming stories); Category 2: Fantasy Adventure Series (Immersive worlds); Category 3: Graphic Novels & Comics (Visual storytelling); Category 4: Classic Literature (Foundational works)."}
</tool_response>
<Thinking> Now let me check specific items from the user's history to see what similar users like. </Thinking>
<tool_call>
{"name": "ItemCF", "arguments": "{"item_id": "739484"}"}
{"name": "ItemCF", "arguments": "{"item_id": "824734"}"}
{"name": "ItemCF", "arguments": "{"item_id": "12347010"}"}
</tool_call>
↪ <tool_response>
{"result": "Category 1: Early Childhood Picture Books (Interactive); Category 2: Graphic Novels and Illustrated Adventures (Visual quests); Category 3: Middle Grade Chapter Books (Mystery); Category 4: Literary Works."}
</tool_response>
↪ <tool_response>
{"result": "Category 1: Children's Adventure Series (Magic); Category 2: Illustrated Picture Books (Animals); Category 3: Artistic Concept Books (Creativity); Category 4: Animal-Centric Stories (Whimsical)."}
</tool_response>
↪ <tool_response>
{"result": "Category 1: Animal Character Stories; Category 2: Series-Based Books; Category 3: Holiday and Seasonal Themes; Category 4: Mystery Stories; Category 5: Everyday Life Lessons."}
</tool_response>
<Thinking> Synthesizing these signals for expansion: Association (Educational to STEM and Multicultural), Scenario (Seasonal to Community and Problem-solving), and Group Commonalities (Fantasy and Graphic Novels). </Thinking>
<JSON>["STEM Education", "Multicultural Literature", "Fantasy Adventure Series", "Graphic Novels", "Classic Adaptations", "Community Themes", "Problem-Solving Stories"]</JSON>
</interest_divergence>

<reflection>
```

```
<Thinking> I have verified the outputs against user data. The ‘User_Profile‘, ‘Historical_Interest‘, and ‘Recent_Interest‘ agents accurately reflect the user’s focus on educational picture books, moral lessons, and specific series like "Little Critter". However, the ‘Interest_Divergence_Reasoning‘ agent is problematic. It proposes speculative expansions into Graphic Novels and Fantasy Adventures, which contradict the user’s consistent preference for simple, realistic educational content. Divergence must remain grounded in the user’s actual reading level and habits. </Thinking>
<JSON> { "correct": "no", "problematic_agent": [ { "agent_name": "Interest_Divergence_Reasoning", "suggestion": "Divergence is too speculative. Suggested genres like Graphic Novels and Fantasy conflict with the user’s established preference for educational picture books. Reasoning should focus on logical extensions (e.g., similar series, interactive formats) rather than broad genre hops." } ] } </JSON>
</reflection>

<interest_divergence>
<Thinking> I need to correct the divergence reasoning based on the Reflector’s feedback. Instead of speculative genre hops (like Graphic Novels), I will focus on logical extensions of the user’s established interests: educational content, moral lessons, and series continuity. </Thinking>
<tool_call>
{"name": "UserCF", "arguments": "{"user_id": "723a2a35783fadad2a4f2e00c5e639d7"}"}
</tool_call>
↪ <tool_response>
{"result": "Category 1: Children’s Picture Books (Heartwarming stories); Category 2: Fantasy Adventure Series (Immersive worlds); Category 3: Graphic Novels & Comics (Visual storytelling); Category 4: Classic Literature (Foundational works)."}
</tool_response>
<Thinking> I will cross-reference specific history items to find safer, more grounded expansions. </Thinking>
<tool_call>
{"name": "ItemCF", "arguments": "[{"item_id": "12347010"}, {"item_id": "824734"}]"}
</tool_call>
↪ <tool_response>
{"result": "Category 1: Popular Children’s Series (Recurring characters); Category 2: Animal Protagonist Stories (Relatable avatars); Category 3: Seasonal and Holiday Themes (Contextual reading); Category 4: Life Lessons and Values (Social skills)."}
</tool_response>
↪ <tool_response>
{"result": "Category 1: Series Adventure Stories (Continuity); Category 2: Fantasy & Imagination (Creative premises); Category 3: Animal Adventures (Humor); Category 4: Educational & Concept Books (Learning)."}
</tool_response>
<Thinking> Synthesizing for grounded divergence: 1. **Association**: From "Moral Lessons" to "Social-Emotional Learning" and "Real-Life Challenges". 2. **Scenario**: From "Seasonal" to "Environmental Awareness" and "Cultural Diversity". 3. **Group**: Focusing on "Series Continuity" and "Interactive Learning" rather than changing genres. </Thinking>
<JSON>["More Character-Driven Series", "Seasonal Educational Stories", "Interactive Learning Books", "Social-Emotional Learning", "Animal Moral Stories", "Simple STEM Concepts", "Cultural Diversity", "Problem-Solving Stories", "Environmental Awareness", "Real-Life Challenges"]</JSON>
</interest_divergence>

<recommend>
<Thinking> I will finalize the recommendation list by verifying user preferences via UserCF and exploring specific item affinities via ItemCF to ensure the ranking aligns with the user’s focus on educational and seasonal content. </Thinking>
<tool_call>
{"name": "UserCF", "arguments": "{"user_id": "723a2a35783fadad2a4f2e00c5e639d7"}"}
</tool_call>
↪ <tool_response>
{"result": "Category 1: Children’s Picture Books & Early Readers (Wholesome life lessons); Category 2: Fantasy Adventure Series (Immersive continuity); Category 3: Comic Books & Graphic Novels (Visual storytelling); Category 4: Classic & Educational Literature (Literary quality)."}
</tool_response>
<Thinking> Checking item-specific collaborative patterns to refine the ranking. </Thinking>
<tool_call>
{"name": "ItemCF", "arguments": "{"item_id": "12347010"}"}
{"name": "ItemCF", "arguments": "{"item_id": "824734"}"}
{"name": "ItemCF", "arguments": "{"item_id": "739484"}"}
</tool_call>
↪ <tool_response>
{"result": "Category 1: Animal-Focused Stories (Relatable characters); Category 2: Series Books for Young Readers (Familiar formats); Category 3: Holiday and Seasonal Themes (Contextual relevance); Category 4: Educational and Value-Based Stories (Moral growth)."}
</tool_response>
↪ <tool_response>
```

```
{"result": "Category 1: Early Reader Chapter Book Series (Mild adventure); Category 2: Picture Book Adventures (Read-aloud); Category 3: Fantasy/Imaginative Worlds (Creative thinking); Category 4: Simple Concept/Educational Books (Basic concepts)."}
</tool_response>
↪ <tool_response>
{"result": "Category 1: Animal-Themed Children's Books (Engaging protagonists); Category 2: Children's Book Series (Collectible); Category 3: Interactive/Sensory Children's Books (Tactile engagement); Category 4: General Children's Fiction (Diverse themes)."}
</tool_response>
<Thinking> Final Ranking Logic: 1. **High Priority**: Items matching "Educational", "Seasonal (Fall/Earth Day)", "Series Continuity (Little Critter style)", and "Interactive" attributes. 2. **Medium Priority**: General children's fiction and fantasy. 3. **Low Priority**: Comic books and graphic novels, as the user shows no history of reading this genre despite the UserCF suggestion. </Thinking>
<JSON>["18850623", "2150618", "1130874", "1292299", "13349511", "36041925", "31767393", "451093", "6261472", "2839512", "18686147", "18762411", "4765806", "2830774", "100580", "14472504", "22017603", "32471983", "8510571", "18686147"]</JSON>
</recommend>
```