LLMs in Recommender System

Wenwei Ban

2025/6/27

STAR Group Paper Reading

Table of Contents

Paper List:

Basic Information

Process-Supervised LLM Recommenders via Flow-guided Tuning

Chongming Gao*
chongminggao@ustc.edu.cn
University of Science and Technology of China
Hefei, China

Mengyao Gao*
mengyao0301@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, China

Chenxiao Fan
simonfan@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, China

Shuai Yuan
syuanaf@connect.ust.hk
Hong Kong University of Science and Technology
Hong Kong, China

Wentao Shi
shiwentao123@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, China

Xiangnan He†
xiangnanhe@gmail.com
MoE Key Lab of BIPC, University of Science and Technology of China
Hefei, China

- *SIGIR '25, 7 citations*

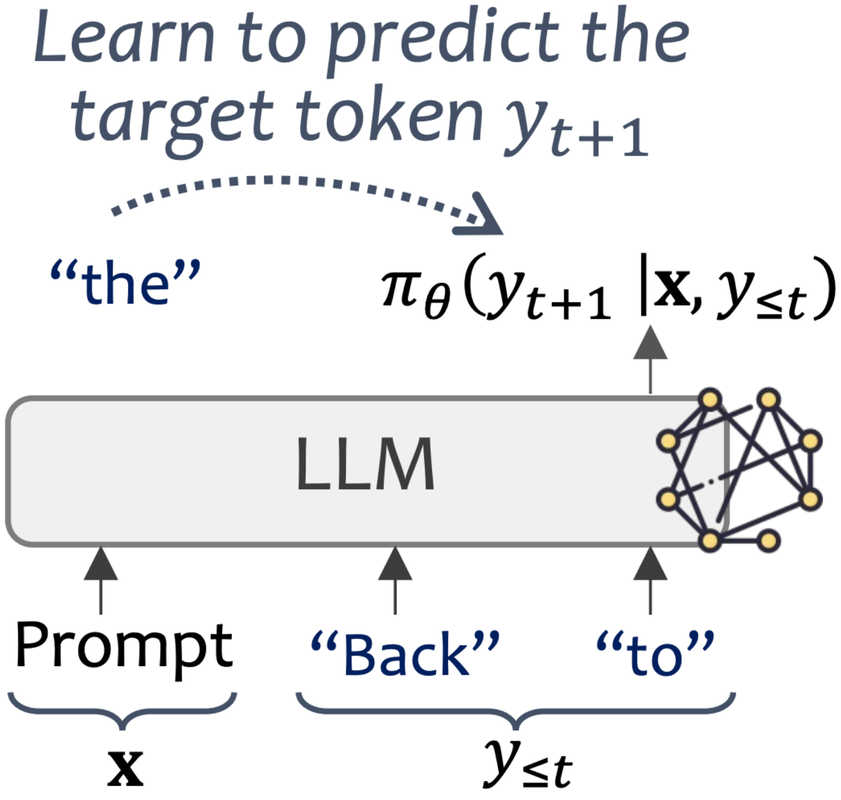
Background

- Methods: LLMs for recommendation systems via **supervised fine-tuning (SFT)**

- Problems:

1. Popularity bias amplification

2. Limited diversity



(a) Supervised fine-tuning

Existing Solutions

Method	Category	Bias-free	Diversity
Reweighting [2]	Modify the SFT learning process	√	×
Multi-stage SFT [3]	Modify the SFT learning process	×	√
RLHF [4,5]	Post-SFT policy optimization	√	×
DPO [6,7]	Post-SFT policy optimization	√	×

Motivation

- Present **Flow-guided** fine-tuning recommender(**Flower**), which replaces SFT with a Generative Flow Network(**GFlowNet**) framework that enacts process supervision through token-level reward propagation.

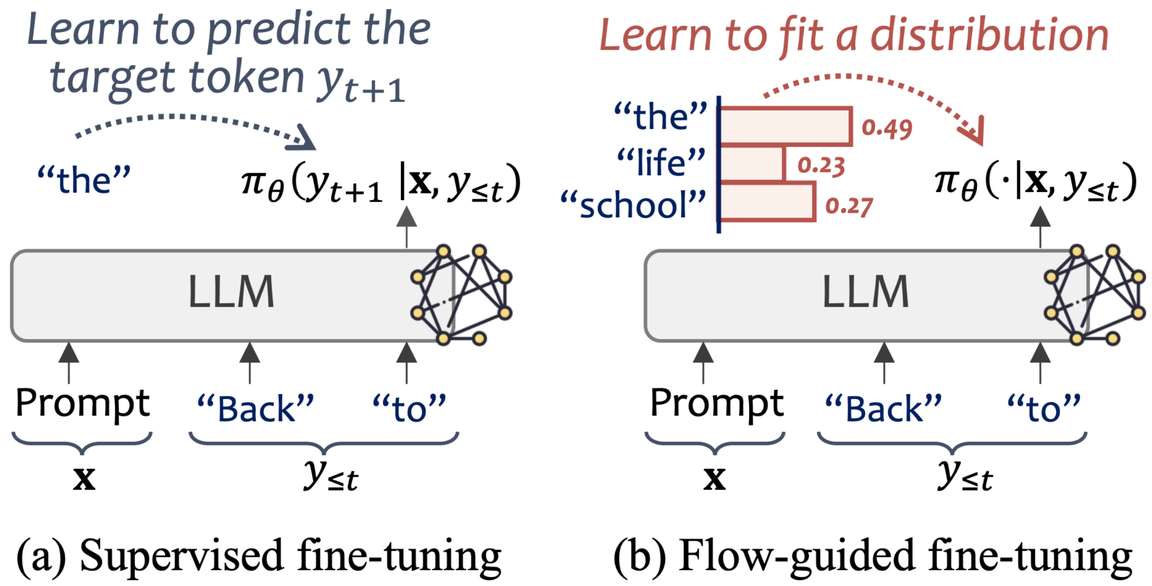
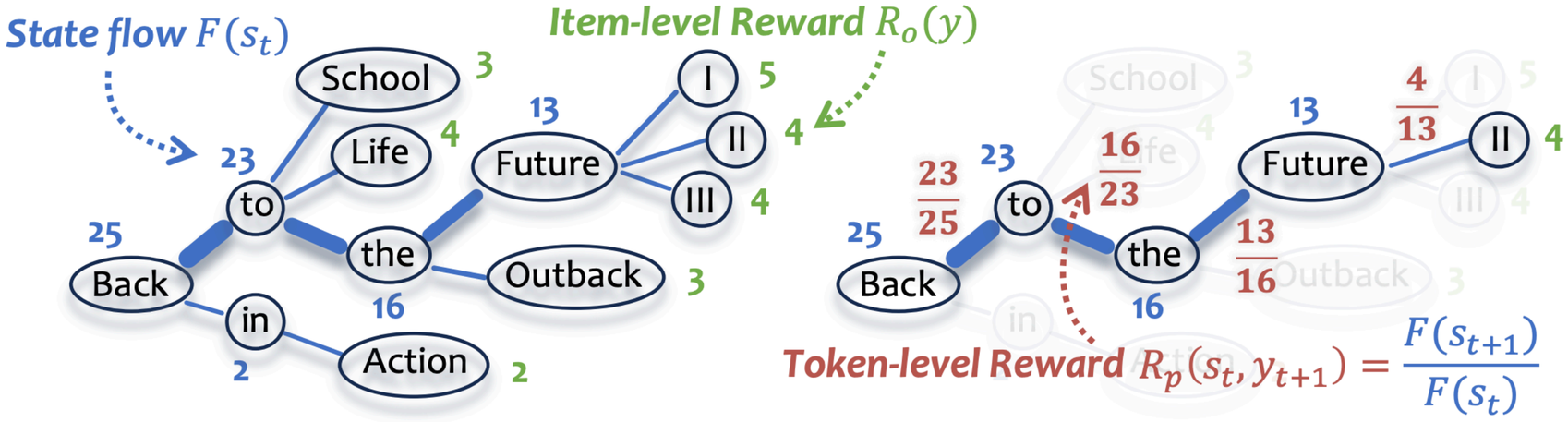


Figure 1: Illustration of two tuning paradigms in LLM-based next-item recommendation tasks

Method: Flower

Movie title	Reward
Back to School	3
Back to Life	4
Back to the Future I	5
Back to the Future II	4
Back to the Future III	4
Back to the Outback	3
Back in Action	2

(a) Item-level outcome rewards



(b) "Flow" in prefix tree of all items

(c) Token-level rewards for process supervision

Figure 3: Illustration of the prefix tree, state flow, item-level rewards, and flow-guided token-level rewards in Flower.

$$\mathscr{L}_R(\tau_{m,n}) = \left(\sum_{t=m}^{n-1} \log \pi_\theta(y_{t+1} \mid x, y_{\le t}) - \sum_{t=m}^{n-1} \log R_p(y_{\le t}, y_{t+1}) \right)^2$$

Reward Setting

- Problem: this reward remains static across all users and does **not account for personalized preferences**.

- Introduce a preference score p_{ui}, which predicts the likelihood of user liking item (can be obtained from any auxiliary model, eg: SASRec).

- Modifying the process reward term $\log R_p(y_{\leq t}, y_{t+1})$ as:

(1) $$\frac{logR_p(y_{\leq t}, y_{t+1})}{p_{ut}}$$

(2) $$log(p_{ui} \cdot R_p(y_{\leq t}, y_{t+1}))$$

Fine-tuning LLMs through Process Rewards

- To fine-tune the policy , we integrate the original SFT loss LSFT from Eq.
- The combined loss function of Flower is formulated as:

$$\mathscr{L}_{Flower} = \mathscr{L}_{SFT} + \lambda \sum_{\tau \in \mathscr{T}} \sum_{0 \le m < n \le T} \mathscr{L}_R\left(\tau_{m,n}\right)$$

- This combined loss preserves the supervised performance of SFT while leveraging GFlowNets to promote diversity and reward-proportionality

Experiments

- Qualitative Visualization

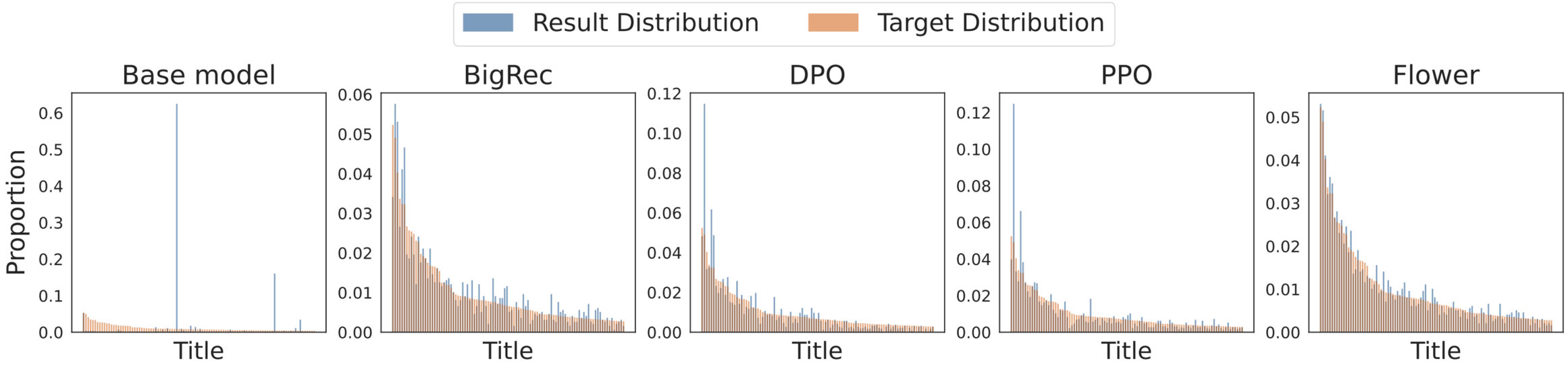


Figure 4: Comparison of the distributions between the target set and the recommended results across 100 movie titles.

- Flower effectively learns the target distribution, capturing titles with varying popularity and mitigating the unfairness observed in other methods

Experiments

- Quantitative Analysis

Table 4: Performance of all methods evaluated in terms of accuracy, fairness, and diversity. The best results are bolded.

	CDs and Vinyl						Video Games						Movies and TV					
	NDCG↑	**HR↑**	**DGU↓**	**MGU↓**	**H↑**	**TTR↑**	**NDCG↑**	**HR↑**	**DGU↓**	**MGU↓**	**H↑**	**TTR↑**	**NDCG↑**	**HR↑**	**DGU↓**	**MGU↓**	**H↑**	**TTR↑**
SASRec	0.0641	0.0851	0.184	0.038	9.188	0.124	0.0369	0.0544	0.167	0.033	8.229	0.050	0.0902	0.1072	0.138	0.032	8.892	0.167
BIGRec	0.0573	0.0715	0.217	0.045	5.900	0.006	0.0326	0.0466	0.151	0.029	7.504	0.004	0.0930	0.1134	0.123	0.028	8.297	0.018
Temp	0.0503	0.0627	0.222	0.044	6.202	0.006	0.0306	0.0444	0.129	0.026	7.307	0.004	0.0852	0.1061	0.139	0.027	8.145	0.018
D3	**0.0812**	**0.0999**	0.355	0.072	7.635	0.013	0.0413	0.0607	0.220	0.041	7.645	0.005	**0.1007**	**0.1225**	0.147	0.033	8.348	0.020
IFairLRS	0.0621	0.0762	0.217	0.045	6.420	0.007	0.0396	0.0568	0.144	0.030	7.699	0.005	0.0957	0.1170	0.159	0.043	8.048	0.015
Flower	0.0700	0.0885	**0.075**	**0.021**	**7.919**	**0.013**	**0.0543**	**0.0799**	**0.108**	**0.023**	**7.750**	**0.005**	0.0959	0.1199	**0.076**	**0.026**	**8.808**	**0.023**

- Compared to baseline methods, Flower achieves optimal fairness and diversity across all dataset

Experiments

- Loss fuction:

$$\mathscr{L}_{Flower} = \mathscr{L}_{SFT} + \lambda \sum_{\tau \in \mathscr{T}} \sum_{0 \le m < n \le T} \mathscr{L}_R\left(\tau_{m,n}\right)$$

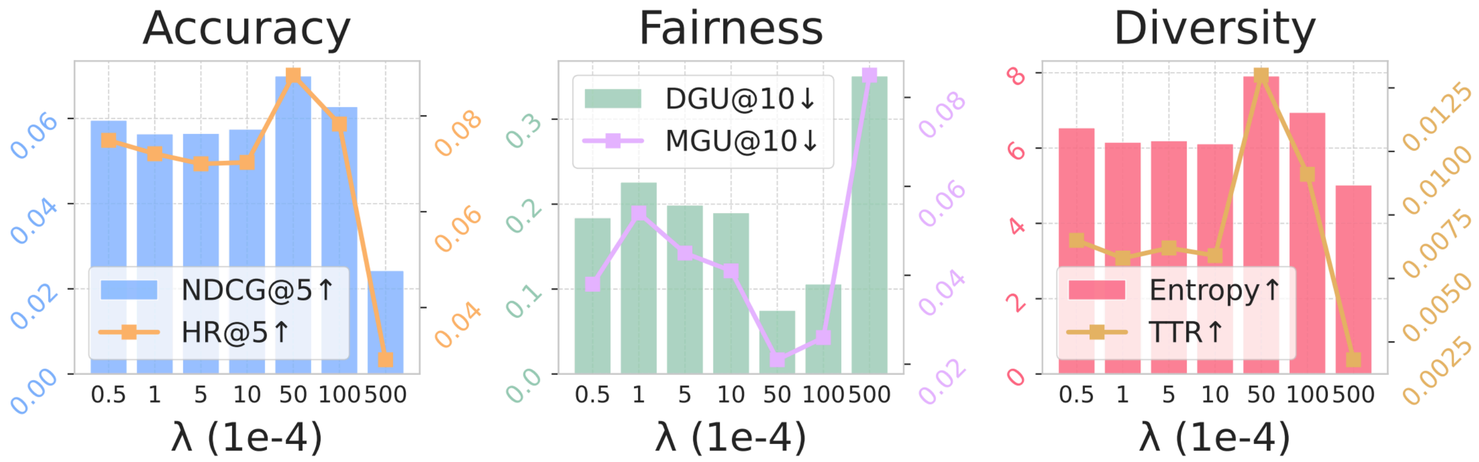


Figure 6: Performance with varying λ on the CDs dataset.

- Accuracy, fairness, and diversity generally exhibit a trend of first improving and then declining, with the best performance observed around = 0.005.

Basic Information

Agentic Feedback Loop Modeling Improves Recommendation and User Simulation

Shihao Cai
University of Science and Technology of China
Hefei, Anhui, China
caishihao@mail.ustc.edu.cn

Jizhi Zhang
University of Science and Technology of China
Hefei, Anhui, China
cdzhangjizhi@mail.ustc.edu.cn

Keqin Bao
University of Science and Technology of China
Hefei, Anhui, China
baokq@mail.ustc.edu.cn

Chongming Gao*
University of Science and Technology of China
Hefei, Anhui, China
chongming.gao@gmail.com

Qifan Wang
Meta AI
Menlo Park, CA, United States
wqfcr@fb.com

Fuli Feng
MoE Key Lab of BIPC, University of Science and Technology of China
Hefei, Anhui, China
fulifeng93@gmail.com

Xiangnan He*
MoE Key Lab of BIPC, University of Science and Technology of China
Hefei, Anhui, China
xiangnanhe@gmail.com

- *SIGIR '25, 7 citations*

Background

Rise of LLM-based Agents in Recommendation

- Large Language Model (LLM)-powered agents are increasingly used in recommender systems due to their **extensive knowledge and reasoning capabilities.**

- Current research focuses on two separate directions:

Recommendation agents (e.g., RecMind, MACRec): Leverage LLMs to improve recommendation accuracy via world knowledge and tool usage.

User simulation agents (e.g., Agent4Rec, RecLLM): Use LLMs to mimic user behaviors (e.g., liking, commenting).

Background

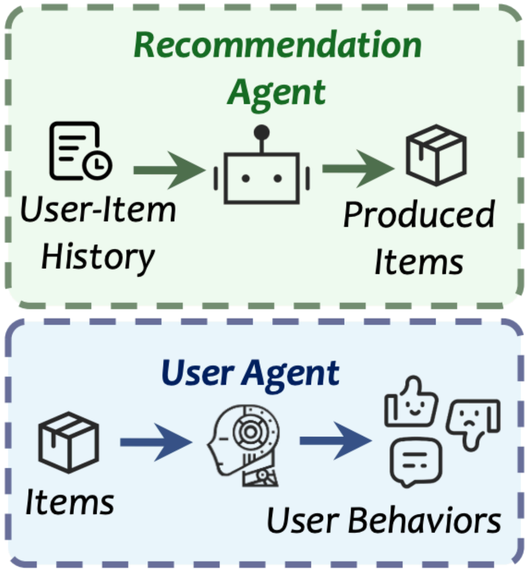
Limitation of Existing Work

- Most studies optimize recommendation agents or user agents **independently**, ignoring the critical **feedback loop** between users and recommenders in real-world scenarios.

- In practice, recommenders and users influence each other:

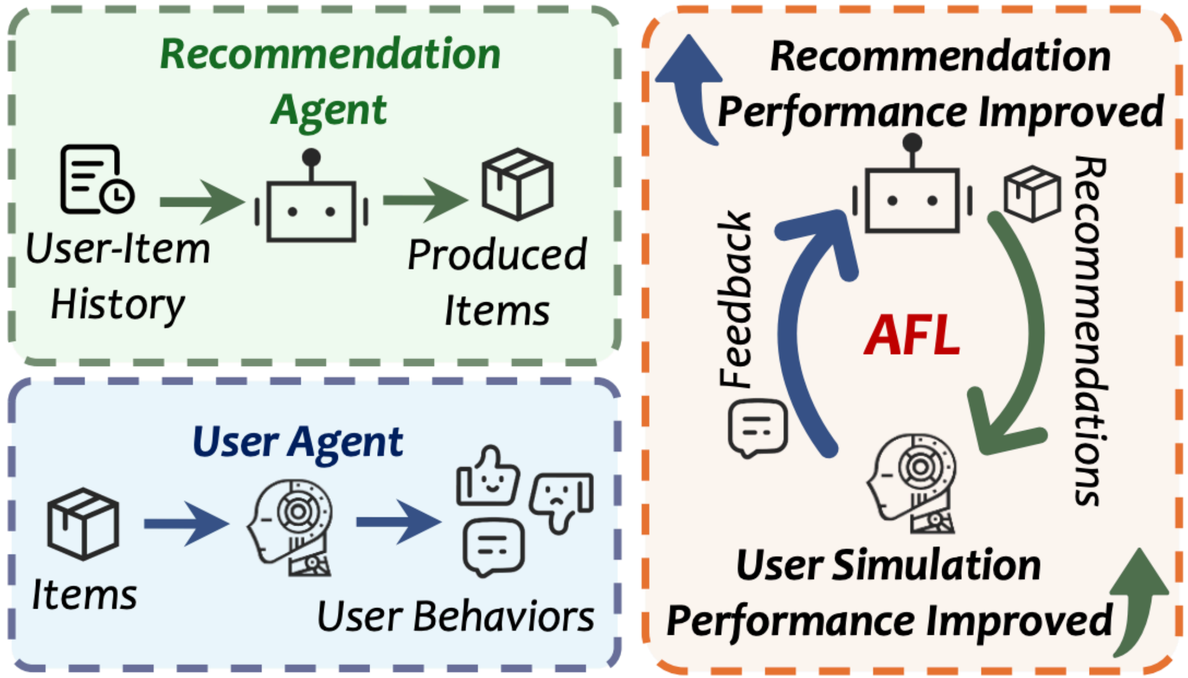
Recommenders help users discover interests.

User feedback refines recommenders' preference understanding.



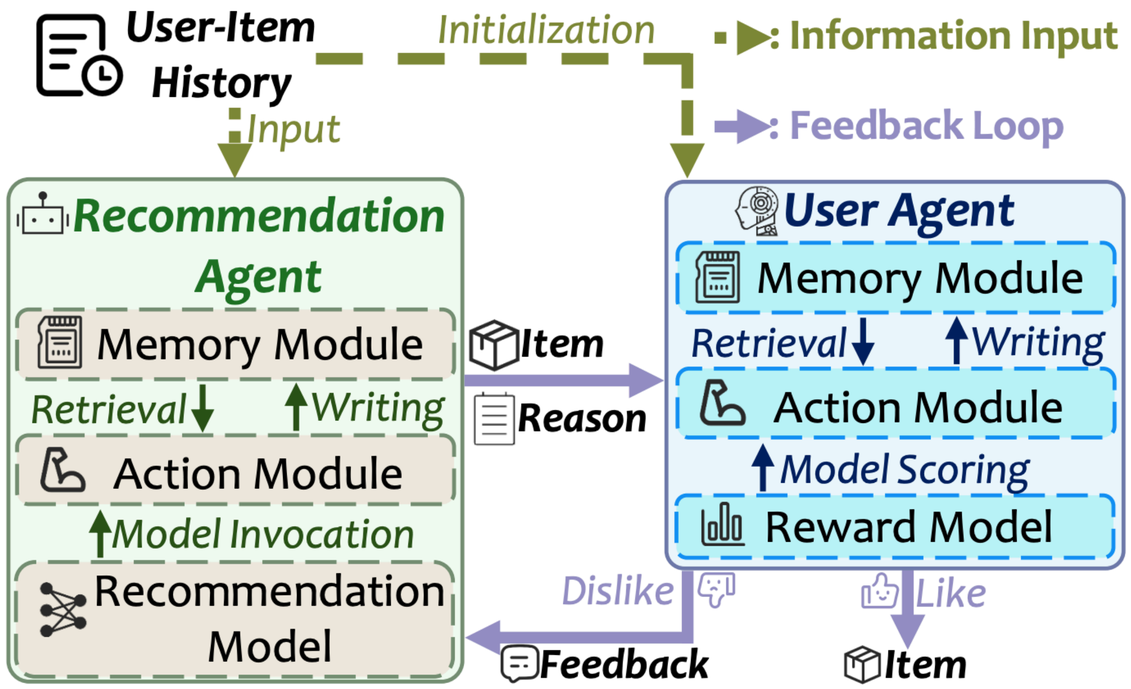
Motivation

- Towards this research gap, we propose a novel framework (AFL) that emphasizes the **feedback loop** process to facilitate the collaboration



AFL Framework: Methodology

- Components:
  - ▶ Recommendation agent (LLM + memory + recommendation model)
  - ▶ User agent (LLM + memory + reward model)
  - ▶ Feedback loop (iterative interaction with memory update)



Memory Template & Prompt Template

Table 1: Memory template and prompt template in the Lastfm dataset for the recommendation agent.

Memory Template
In round {}, the music artist you recommended is {}. The reason you gave for the recommendation is: {}. The reason the user provided for not considering this to be the best recommendation is: {}.
Prompt Template
You are a music artist **recommendation system**. Refine the user's listening history to **predict the most likely music artist** he/she will listen to next from the candidate list. Here is the **history of communication** between you and the user: {}. **Another recommendation model** has suggested a music artist for your reference: {}. **Some useful tips:** 1. You need to first **give the reasons**, and then provide the recommended music artist. 2. The recommended music artist must be on the candidate list. You must follow this output format: **Reason**: <your reason example> **Item**: <item example>

Table 2: Memory template and prompt template in the Lastfm dataset for the user agent.

Memory Template
In round {}, the recommended music artist is {}. The reason given by the recommendation system is: {} The reason you provided for not considering this the best recommendation is {}
Prompt Template
As a **music listener**, you've listened to the following music artists: {}. Now, a recommendation system has recommended a music artist to you from a list of music artist candidates, and has provided the reason for the recommendation. **Determine if this recommended music artist is the most preferred option** from the list of candidates based on your personal tastes and previous listening records. Here is the **history of communication** between you and the recommendation system: {} What's more, a **reward model** scores the music artist based on its relevance to your historical listening records: {} **Some useful tips:** 1. You need to first **give the reasons**, and then decide whether or not the recommended music artist is the most preferred one on the candidate list for you. 2. **Summarize your own interests** based on your historical listening records to make a judgment. 3. You can **refer to the score** given by the reward model. You must follow this output format: **Reason**: <your reason example> **Decision**: <yes or no>

Experiments

- In this section, we conduct experiments to answer the following research questions (RQ):
- RQ1: Can AFL **enhance performance** in both the recommendation task and the user behavior simulation task?
- RQ2: What are the effects of **the key components** of AFL?

Experimental Setup

- LLM: GPT-4o-mini
- Reward Model: SASRec
- Datasets: Lastfm, Steam, and MovieLens
- Dataset Splitting Strategy：8(train):1(valid):1(test)

Recommendation Performance (RQ1)

Table 4: The recommendation performance of AFL compared with "Base Model" and "Rec Agent". Bold indicates the best performance. The maximum number of feedback loop iterations for AFL is 4.

Type	Model	Lastfm			Steam			MovieLens		
		Base Model	Rec Agent	AFL	Base Model	Rec Agent	AFL	Base Model	Rec Agent	AFL
Traditional	SASRec	0.2869	0.3197	**0.3770**	0.3800	0.3900	**0.4100**	0.4105	0.4105	**0.4316**
	GRU4Rec	0.2787	0.3114	**0.3770**	0.3750	0.3850	**0.4100**	0.4526	0.4526	**0.4632**
	Caser	0.2705	0.2705	**0.3443**	0.4200	0.4150	**0.4500**	0.3789	0.3895	**0.4000**
LLM-based	MoRec	0.1639	0.2131	**0.3115**	0.4100	0.4200	**0.4250**	0.3158	0.3158	**0.3474**
	Llama3-8B	0.2131	0.2541	**0.2869**	0.1800	0.2250	**0.3000**	0.1368	0.1368	**0.1684**
	GPT-4o-mini	0.3607	0.3607	**0.3770**	0.3350	0.3400	**0.3500**	0.1368	0.1368	**0.1579**
	LLaRA	0.4426	0.4426	**0.4836**	0.4650	0.4650	**0.4750**	0.4842	0.4842	**0.4947**

- AFL can improve the performance of recommendation agents equipped with various base models

User Simulation Performance (RQ1)

Table 5: The user simulation performance of AFL compared with "Reward Model" and "User Agent". Bold results indicate the best results. The maximum number of feedback loop iterations for AFL is 4.

$1:k$	Method	Lastfm			Steam			Movielens		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
1:1	Reward Model	0.6667	0.0533	0.0988	0.7826	0.6800	0.7277	0.6929	0.3800	0.4908
	User Agent	0.8155	0.3467	0.4865	0.8031	**0.6933**	0.7422	0.7049	0.5133	0.5941
	AFL	**0.8504**	**0.5000**	**0.6297**	**0.8501**	0.6700	**0.7494**	**0.7065**	**0.5500**	**0.6185**
1:3	Reward Model	0.4167	0.0571	0.1005	0.5791	0.7133	0.6393	0.5179	0.3077	0.3860
	User Agent	0.5910	0.3571	0.4452	0.6323	0.7067	0.6674	0.5114	**0.4800**	0.4952
	AFL	**0.7343**	**0.4286**	**0.5412**	**0.6815**	**0.7267**	**0.7034**	**0.8107**	0.4667	**0.5924**
1:9	Reward Model	0.1667	0.0667	0.0952	0.3408	0.7667	0.4718	0.3397	0.2667	0.2988
	User Agent	0.2356	0.2667	0.2501	0.3682	**0.8167**	0.5076	0.2313	**0.5000**	0.3163
	AFL	**0.3705**	**0.4286**	**0.3974**	**0.4303**	**0.8167**	**0.5636**	**0.4410**	0.4333	**0.4371**

- AFL can improve the performance of user simulation agents

Impact of Key Components (RQ2)

Table 7: Comparison of HitRatio@1 under different settings. Bold results indicate the best results.

Method	Lastfm	Steam	Movielens
AFL	**0.3770**	**0.4100**	**0.4316**
AFL w/o Rec Model	0.3525	0.3950	0.4000
AFL w/o Reward Model	0.3689	0.4000	0.4211
AFL w/o Both	0.3443	0.3250	0.2105

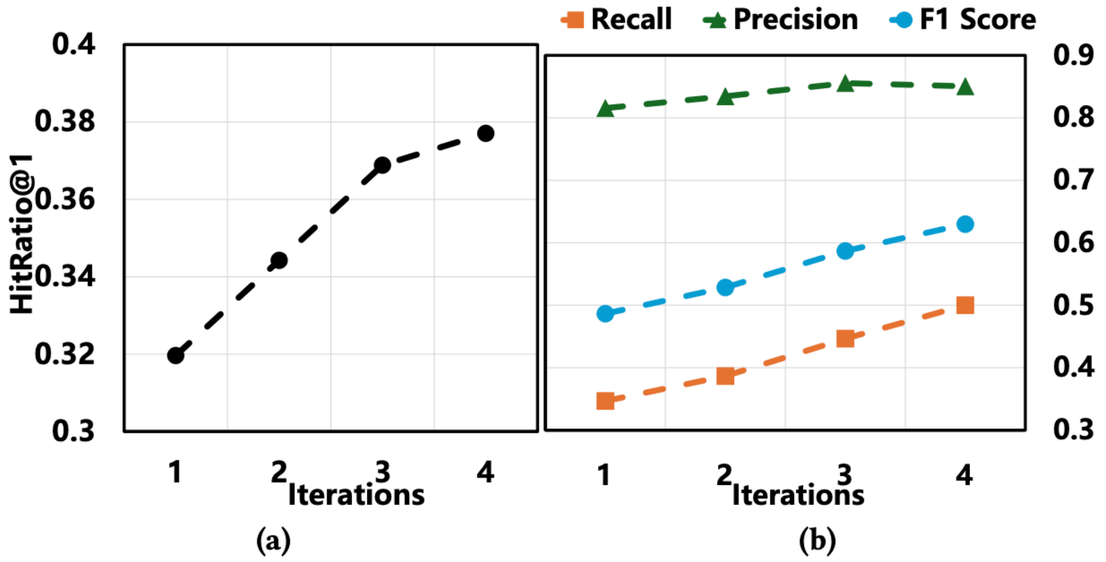


Figure 4: (a) Recommendation performance with increased iterations. (b) User simulation performance with increased iterations. $1 : k$ **is set to** $1 : 1$.

Thanks!!