ARROW: Adaptive Reasoning for LLM-based Recommendation with Explainability

Woo-Seong Yun
Chung-Ang University
Seoul, Republic of Korea
dntjd0804@cau.ac.kr

Min-Seong Kim
Chung-Ang University
Seoul, Republic of Korea
maruniverse@cau.ac.kr

Yoon-Sik Cho*
Chung-Ang University
Seoul, Republic of Korea
yoonsik@cau.ac.kr

Abstract

The integration of Large Language Models (LLMs) has led to substantial advancements in recommender systems (RS) by leveraging their vast knowledge and reasoning abilities. However, the semantic gap between the linguistic knowledge of LLMs and the collaborative patterns in RS hinders their effective fusion. This issue results in a fundamental limitation where models, despite achieving high prediction accuracy, are unable to provide coherent rationales justifying their recommendations. In this paper, we propose ARROW (**A**daptive **R**easoning for LLM-based **R**ecommendati**O**n **W**ith explainability), a novel framework that effectively elicits the intrinsic reasoning capabilities of LLMs to bridge this semantic gap. ARROW is carefully designed to guide the model in generating an explicit reasoning process for its recommendation decisions using chain-of-thought prompting. Furthermore, we introduce the Adaptive Reasoning Modulator, which quantifies the uncertainty of the reasoning process and adaptively adjusts its weight to maximize the model's reasoning efficacy. Our extensive experiments demonstrate that ARROW achieves significant performance improvements over strong baseline models while providing human-interpretable explanations. Our code is available at https://github.com/yunwooseong/ARROW.

CCS Concepts

• **Information systems → Recommender systems**.

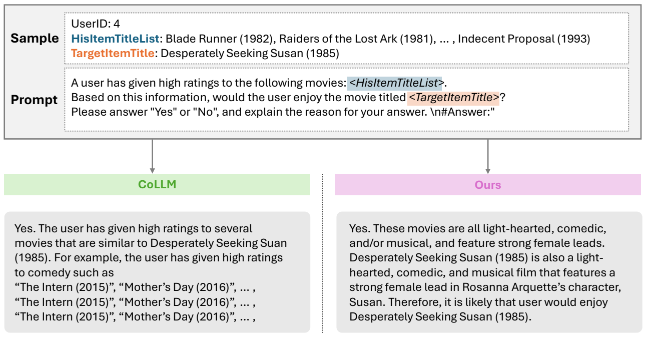
Keywords

Recommender System; Large Language Models; LLM Reasoning

ACM Reference Format:
Woo-Seong Yun, Min-Seong Kim, and Yoon-Sik Cho. 2026. ARROW: Adaptive Reasoning for LLM-based Recommendation with Explainability. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining (WSDM '26), February 22–26, 2026, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3773966.3779396

1 Introduction

Recommender systems (RS) have emerged as a pivotal technology for mitigating information overload and enhancing user experiences across various online platforms. Collaborative Filtering (CF) [16, 21], which models user preferences based on user-item interactions, has been widely used in RS. With the remarkable advancements in Large Language Models (LLMs), there has been growing interest in leveraging their rich knowledge and reasoning capabilities for RS.



UserID: 4
HisItemTitleList: Blade Runner (1982), Raiders of the Lost Ark (1981), ... , Indecent Proposal (1993)
TargetItemTitle: Desperately Seeking Susan (1985)

A user has given high ratings to the following movies: *<HisItemTitleList>*.
Based on this information, would the user enjoy the movie titled *<TargetItemTitle>*?
Please answer "Yes" or "No", and explain the reason for your answer. \n#Answer:"

Yes. The user has given high ratings to several movies that are similar to Desperately Seeking Suan (1985). For example, the user has given high ratings to comedy such as
"The Intern (2015)", "Mother's Day (2016)", ... ,
"The Intern (2015)", "Mother's Day (2016)", ... ,
"The Intern (2015)", "Mother's Day (2016)", ... ,

Yes. These movies are all light-hearted, comedic, and/or musical, and feature strong female leads. Desperately Seeking Susan (1985) is also a light-hearted, comedic, and musical film that features a strong female lead in Rosanna Arquette's character, Susan. Therefore, it is likely that user would enjoy Desperately Seeking Susan (1985).

Figure 1: Case study of CoLLM and the proposed ARROW on ML-1M dataset.

Early studies [1, 5, 7, 13] attempted to align LLMs with recommendation tasks by converting recommendation data into natural language formats. CoLLM [27] enhances this mechanism by providing pre-trained collaborative embedding as special tokens within the prompt, combining the LLM's prior knowledge with collaborative knowledge. While these approaches have shown some promise, a fundamental challenge remains. This challenge, which we term the *semantic gap*, stems from the inherent heterogeneity between LLM's pre-trained linguistic knowledge and collaborative patterns in interaction data. As recent literature [10, 22] highlights, this semantic gap hinders the effective fusion of the two knowledge sources. This heterogeneous knowledge fusion results in shallow integration, hindering LLMs from fully utilizing their model capacity for recommendation tasks [29].

To bridge this semantic gap, recent literature [2, 3, 14, 25] highlights the importance of a reasoning process that justifies their recommendations. In other words, it is essential to leverage the reasoning capabilities of LLMs to fuse these heterogeneous knowledge. However, existing LLM-based models [1, 18, 19, 26, 27] are optimized for simplistic outputs (e.g., such as "Yes/No" predictions or single rating scores), which fail to provide the rationale behind a recommendation. Even models designed to generate explanations are limited in their functionalities, and merely summarize or extract snippets from review texts [14]. As illustrated in Figure 1, CoLLM [27] correctly predicts the outcome but fails to provide a rational explanation for its reasoning, often producing hallucinations such as repeating movies that are not in the user's history.

In this paper, we propose ARROW (**A**daptive **R**easoning for LLM-based **R**ecommendati**O**n **W**ith explainability), a novel framework that elicits the intrinsic reasoning capabilities of LLMs to

[*] Corresponding author.

WSDM '26, Boise, ID, USA

ACM ISBN 979-8-4007-2292-9/2026/02
https://doi.org/10.1145/3773966.3779396

bridge the semantic gap. ARROW is carefully designed to have the model generate an explicit reasoning process employing chain-of-thought (CoT) approach [23]. This reasoning process is directly integrated into the model's optimization, enabling the model to achieve a deeper understanding of the user. Furthermore, we introduce the Adaptive Reasoning Modulator (ARM), which maximizes efficacy by adaptively adjusting the training weight based on the model's reasoning uncertainty. A side advantage of ARROW is its explainability, which makes its decision-making process human-understandable. Our extensive experimental results demonstrate that ARROW outperforms existing strong baselines.

2 Preliminaries

2.1 Problem Definition

The click-through rate prediction task aims to predict whether a user will click on a specific item based on their historical interactions. Given a user-item interaction dataset $\mathcal{D} = \{(u, i, y)\}$, where each tuple consists of user $u \in \mathcal{U}$, an item $i \in \mathcal{I}$, and a label $y \in \{0, 1\}$ indicating click occurrence. User interaction history and target items are converted into textual representations (e.g., titles) for model input. Our objective is to predict whether user u will click on the target item by leveraging both the textual embeddings $\mathbf{e}_t$ obtained through LLMs and the user/item embeddings $\mathbf{e}_u$ and $\mathbf{e}_i$ derived from pre-trained CF models.

2.2 Pretrained CF model

The CF model aims to extract collaborative embeddings $(\mathbf{e}_u, \mathbf{e}_i)$ for each user u and item i from the user-item interaction data $\mathcal{D}$. This process can be formulated as follows:

$$\mathbf{e}_u = f_\psi(u; \mathcal{D}), \quad \mathbf{e}_i = f_\psi(i; \mathcal{D}), \tag{1}$$

where f_ψ represents a CF model parameterized by ψ. Each user u and item i are mapped to d_1-dimensional vector representations ($\mathbf{e}_u, \mathbf{e}_i \in \mathbb{R}^{d_1}$). The obtained collaborative embeddings, which summarize user behavioral patterns, are integrated with textual information within the LLM to effectively model user-item relationship.

2.3 CoLLM

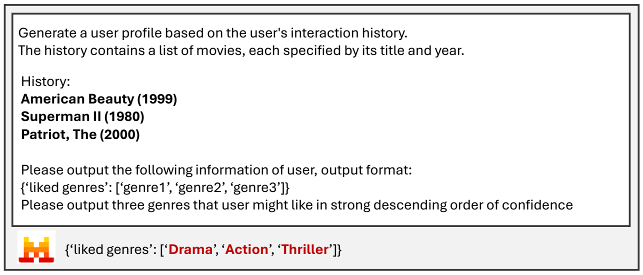
CoLLM [27] is a representative framework for integrating the prior knowledge of LLMs with collaborative knowledge. This model adopts a collaborative information encoding (CIE) module and a two-stage training strategy to enable seamless integration of these two heterogeneous knowledge sources.

CIE Module. This module maps collaborative embeddings $\mathbf{e}_u$ and $\mathbf{e}_i$ provided in the prompt into the LLM's token embedding space. This mapping is performed by a layer g_ϕ, which is a multilayer perceptron (MLP) parameterized by ϕ:

$$\mathbf{h}_u = g_\phi(\mathbf{e}_u), \quad \mathbf{h}_i = g_\phi(\mathbf{e}_i). \tag{2}$$

Here, $\mathbf{h}_u, \mathbf{h}_i \in \mathbb{R}^{d_2}$ are the final collaborative representations used by the LLM for prediction.

Two-Stage Training. In the first stage, the LLM is adapted to the collaborative patterns by fine-tuning via LoRA with only textual information. Subsequently, in the second stage, the LLM and LoRA parameters are frozen, and only the CIE module is trained to map the collaborative information into the LLM's embedding space.



Generate a user profile based on the user's interaction history.
The history contains a list of movies, each specified by its title and year.

History:
American Beauty (1999)
Superman II (1980)
Patriot, The (2000)

Please output the following information of user, output format:
{'liked genres': ['genre1', 'genre2', 'genre3']}
Please output three genres that user might like in strong descending order of confidence

{'liked genres': ['**Drama**', '**Action**', '**Thriller**']}

Figure 2: Prompt used for generating ground-truth genre.

3 Proposed Method

In this section, we introduce ARROW (**A**daptive **R**easoning for LLM-based **R**ecommendati**O**n **W**ith explainability), which are carefully designed to elicit the LLM's reasoning capabilities. Our approach guides the LLM to explicitly infer a user's latent tastes through CoT prompting. We begin with Section 3.1, which describes how to generate ground-truth genres for reasoning.

3.1 Genre Generation

In order to ensure effective reasoning, we generate ground-truth genre labels based on user interaction history. We employ Mistral-Nemo-Instruct to extract preferred genres (e.g., "Drama") following the generation strategy in [24]. The ground-truth label for user u is formalized as a multi-hot binary vector, $\mathbf{g}_u \in \{0, 1\}^{|G|}$. Each element in this vector is set to 1 if the corresponding genre is included in the previously generated list of preferred genres, and 0 otherwise. The detailed prompt is presented in Figure 2. Interestingly, providing additional movie genre information yield no significant performance difference; we defer to Section 4.2.3.

3.2 Chain-of-Thought Prompt Construction

We design a CoT prompt to fully leverage the intrinsic reasoning capabilities of LLMs. Following a widely adopted technique [23] for eliciting explicit reasoning, we incorporate step-by-step instruction. Our prompt template is as follows:

Prompt Template

#Question: A user has given high ratings to the following movies: <*ItemTitleList*>. Additionally, we have information about the user's preferences encoded in the feature <*UserID*>. Using all available information, let's think step-by-step: First, infer the genres the user is most likely to enjoy. Then, make a prediction about whether the user would enjoy the movie titled <*TargetItemTitle*> with the feature <*TargetItemID*>? Answer with "Yes" or "No". #Answer:

In this template, $\langle ItemTitleList \rangle$ consists of the titles of items in the user's interaction history, and $\langle TargetItemTitle \rangle$ represents the title of the target item to be predicted. The $\langle UserID \rangle$ and $\langle TargetItemID \rangle$ fields function as special token positions where the collaborative embeddings $\mathbf{e}_u$ and $\mathbf{e}_i$ are injected. This prompt can effectively leverage the model's reasoning capabilities by guiding the LLM to explicitly perform reasoning processes (e.g., genre prediction) before making the recommendation.

3.3 Reasoning Through Genre Prediction

To effectively elicit the reasoning capabilities of LLMs, we define genre prediction as a multi-label classification problem. We add genre-specific special tokens (e.g., [GENRE DRAMA]) to the tokenizer and extend the model's embedding matrix. Through this process, we can obtain both final recommendation predictions and inferred genre information simultaneously. The CoT prompt is encoded as a hybrid embedding sequence $\mathbf{E}$ combining textual and collaborative embeddings:

$$\mathbf{E} = [\mathbf{e}_{t_1}, ..., \mathbf{e}_{t_k}, \mathbf{h}_u, \mathbf{e}_{t_{k+1}}, ..., \mathbf{h}_i, ..., \mathbf{e}_{t_K}], \tag{3}$$

where $\mathbf{e}_{t_k} \in \mathbb{R}^{d_2}$ is the text token embedding in the LLM, and $\mathbf{h}_u, \mathbf{h}_i \in \mathbb{R}^{d_2}$ are the mapped collaborative embeddings.

Feeding the sequence $\mathbf{E}$ into the LLM yields two primary outputs: the final recommendation and the genre prediction:

$$\hat{y}_{\text{rec}}, \hat{y}_{\text{genre}} = \text{LLM}_{\hat{\Theta}+\Theta'+\phi}(\mathbf{E}), \tag{4}$$

where $\hat{\Theta}$ represents the frozen LLM parameters, Θ' represents the trainable LoRA parameters, and ψ indicates the parameters of the alignment module. The training process consists of two stages: in the first stage, $\hat{\Theta}$ is optimized, while in the second stage, ϕ is optimized. $\hat{y}_{\text{rec}}$ represents the final recommendation prediction logits for the "yes" response, while $\hat{y}_{\text{genre}}$ denotes the prediction logits of the top 3 genres with the highest logits among all genres.

3.4 Objective Function

We now consider the optimization of model parameters. First, the recommendation objective $\mathcal{L}_{\text{rec}}$ consists of binary cross-entropy loss for the "Yes/No" prediction. To ensure stability, this loss is computed directly from the model's output logits:

$$\mathcal{L}_{\text{rec}} = \text{BCE}(\hat{y}_{\text{rec}}, y), \tag{5}$$

where $\hat{y}_{\text{rec}}$ represents the recommendation predictions, and $y \in \{0, 1\}$ denotes the ground-truth label.

While simply generating genres is also a promising approach (see Section 4.2.2), we incorporate genre prediction into the training process to effectively elicit the reasoning capabilities of LLMs. The genre loss is defined as follows:

$$\mathcal{L}_{\text{genre}} = \text{BCE}(\hat{y}_{\text{genre}}, \mathbf{g}_u), \tag{6}$$

where $\hat{y}_{\text{genre}}$ represents the genre prediction logits, and $\mathbf{g}_u$ denotes the actual genre preferences of user u.

The total loss function is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + w \cdot \mathcal{L}_{\text{genre}}, \tag{7}$$

where w is the weight parameter used for genre loss.

By jointly optimizing this objective function, ARROW guides the LLM to not only predict user's choices but also form explicit and verifiable reasoning based on user's history. This approach can effectively bridge the gap between collaborative patterns and LLM's knowledge, enhancing the recommedation explainability.

Table 1: Statistics of the processed datasets.

Dataset	#Train	#Valid	#Test	#Users	#Items
ML-1M	33,891	10,401	7,331	839	3,256
Amazon-Book	727,468	25,747	25,747	22,967	34,154

3.5 Adaptive Reasoning Modulator (ARM)

A key challenge in multi-task learning is balancing the contributions of different tasks. Considering not all recommendation tasks require the same level of reasoning, we introduce the Adaptive Reasoning Modulator (ARM), which adaptively adjusts the weight of the genre loss based on the model's predictive uncertainty. We first quantify this uncertainty using Shannon Entropy. We obtain a probability distribution $\mathbf{p_u}$, over the genres, by applying a sigmoid function to the genre logits $\hat{y}_{\text{genre}}$. The entropy for the user u is then calculated over all $|G|$ genres as:

$$H(\mathbf{p}_u) = -\sum_{j=1}^{|G|}(p_{u,j}\log(p_{u,j}) + (1 - p_{u,j})\log(1 - p_{u,j})). \tag{8}$$

The ARM uses a normalized form of this entropy,

$$w_{arm} = \sqrt{\text{mean}_{batch}(H(\mathbf{p}_u)^2)}, \tag{9}$$

as a direct, sample-specific weight for the auxiliary loss. We intentionally use the root mean square because it amplifies the influence of samples with high uncertainty by squaring the entropy scores. This ensures that even if only a few challenging samples exist within a batch, their high uncertainty is not diluted by the majority of easier samples. Consequently, when the model's prediction for a user's genre preferences is uncertain (high entropy), the ARM increases the weight to guide the model to concentrate on the reasoning task.

4 Experiments

4.1 Experimental Setup

Following the literature [18, 19, 26, 27], we evaluate our model on ML-1M [8] and Amazon-Book [9]. The statistics of each dataset are presented in Table 1. We employ AUC, UAUC [17], and NDCG as evaluation metrics. UAUC measures personalized performance by averaging the AUC for each user, while NDCG evaluates the accuracy of the recommendation ranking. We compare against traditional CF models (MF [15], SASRec [12]) and LLM-based models (ICL [6], Prompt4NR [28], TALLRec [1], CoLLM [27], BinLLM [26], CoRA [18]). All baseline implementations follow the configurations reported in their original papers. For LLM-based models, we use Vicuna-7B [4] as the backbone and finetune with LoRA [11], using the AdamW [20] optimizer. All experiments are conducted on two NVIDIA A100 80GB GPUs.

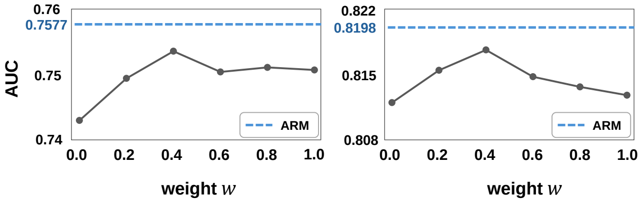
4.2 Experimental Results

Our experiments are designed to answer the following three research questions. **RQ1:** How does ARROW perform on standard benchmarks? **RQ2:** How effective is ARM compared to tuneable w values. **RQ3:** What is the impact of incorporating genre metadata on the ARROW's performance?

*4.2.1 Overall Performance (**RQ1**).* To validate the performance of ARROW, we compare it against strong baseline models on two benchmark datasets. Table 2 highlights that ARROW consistently achieved the best performance across all evaluation metrics. Notably, ARROW achieves significant improvement in the UAUC metric, a key indicator of personalization. This suggests that ARROW's explicit reasoning process leads to a deeper understanding of each

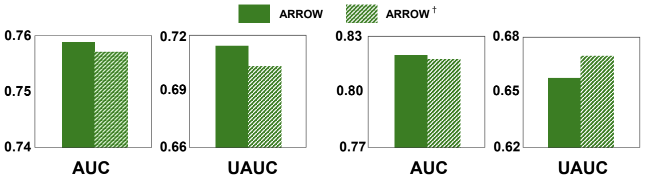
Table 2: Comparison between the proposed ARROW and the baselines. Numbers in **bold** is the best performance and the second best results for each dataset are underlined. All performance improvements are statistically significant based on paired t-tests ($p \leq 0.05$).

Method		ML-1M			Amazon-Book		
		AUC ↑	UAUC ↑	NDCG ↑	AUC ↑	UAUC ↑	NDCG ↑
Traditional CF	MF [15] Computer'09	0.6482	0.6361	0.8447	0.7119	0.5554	0.8194
	SASRec [12] ICDM'18	0.7055	0.6885	0.8612	0.6829	0.5800	0.8244
LLM-based	ICL [6] RecSys'23	0.5320	0.5268	0.8102	0.4820	0.4856	0.7917
	Prompt4NR [28] SIGIR'23	0.7071	0.6739	0.8541	0.7224	0.5881	0.8346
	TALLRec [1] RecSys'23	0.7072	0.6743	0.8578	0.7209	0.5814	0.8361
	CoLLM (MF) [27] TKDE'25	0.7295	0.6798	0.8658	0.8107	0.6029	0.8557
	BinLLM [26] ACL'24	0.7412	0.6951	0.8747	0.8186	0.6338	0.8580
	CoRA [18] AAAI'25	0.7410	0.7061	0.8728	0.8109	0.5975	0.8413
	Arrow	**0.7577**	**0.7146**	**0.8792**	**0.8198**	**0.6576**	**0.8653**



(a) ML-1M (b) Amazon-Book

Figure 3: Effect of genre weight w on the AUC.



(a) ML-1M (b) Amazon-Book

Figure 4: Comparison of Different Genre Generation Approaches.

user's unique preferences. This quantitative performance improvement is further corroborated by our qualitative analysis. As seen in Figure 1, in contrast to CoLLM, ARROW provides a logical and clear rationale for its recommendation results.

*4.2.2 Effectiveness of the ARM (**RQ2**).* We compare dynamic w_{arm} against variants using a static weight w in Equation 7 for the genre loss, where $w \in \{0.0, 0.2, ..., 1.0\}$. The results in Figure 3 offer three key insights. First, even when the reasoning results are not incorporated into the training loss (w=0.0), the reasoning process by itself already brings performance improvement. Second, integrating the genre loss with a static weight (w>0) leads to even greater performance gains, which demonstrates the benefit of explicitly training on the reasoning task. Finally, ARM consistently outperforms all static-weight variants and achieves the best performance. These results highlight the effectiveness of ARM's core mechanism: adaptively adjusting the reasoning loss based on uncertainty.

*4.2.3 Analysis on Genre Generation Strategy (**RQ3**).* We investigate the model's reliance on explicit metadata by comparing two different ground-truth genre generation methods. The standard ARROW model generates ground-truth genres using only item titles via the prompt presented in Figure 2. In contrast, the comparison model, ARROW†, generates them using both item titles and their corresponding genre information. Figure 4 presents the performance comparison results on both datasets. Interestingly, the standard ARROW outperforms ARROW† across most metrics except for UAUC on the Amazon-Book dataset. This can be interpreted as explicit genre information constraining the model's reasoning to predefined categories, whereas title-based reasoning more effectively captures users' complex preferences. These results demonstrate the robustness of our model without reliance on external metadata.

5 Conclusion

In this work, we discuss the semantic gap that arises from the heterogeneity between the prior knowledge of LLMs and the collaborative patterns inherent in recommender systems. To address this challenge, we propose ARROW, a novel framework designed to mitigate this gap by eliciting the intrinsic reasoning capabilities of LLMs. ARROW guides the model in generating an explicit reasoning process for its recommendation decisions using chain-of-thought prompting. This reasoning process is directly integrated into the model's optimization, enabling the model to achieve a deeper understanding of the user. Furthermore, we introduce the Adaptive Reasoning Modulator (ARM), which quantifies the model's reasoning uncertainty and adaptively adjusts the learning process to maximize efficacy. Extensive experimental results demonstrate that ARROW achieves enhanced performance even without additional metadata, while providing human-interpretable explanations.

6 Acknowledgments

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)] and partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00419201).

References

[1] 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*. 1007–1014.

[2] Millennium Bismay, Xiangjue Dong, and James Caverlee. 2025. ReasoningRec: Bridging Personalized Recommendations and Human-Interpretable Explanations through LLM Reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, Luis Chiruzzo, Alan Ritter, and Lu Wang (Eds.). Association for Computational Linguistics, Albuquerque, New Mexico, 8132–8148. doi:10.18653/v1/2025.findings-naacl.454

[3] Jiaju Chen, Chongming Gao, Shuai Yuan, Shuchang Liu, Qingpeng Cai, and Peng Jiang. 2025. Dlcrec: A novel approach for managing diversity in llm-based recommender systems. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*. 857–865.

[4] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

[5] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1126–1132.

[6] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1126–1132.

[7] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).

[8] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[9] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[10] Minjie Hong, Yan Xia, Zehan Wang, Jieming Zhu, Ye Wang, Sihang Cai, Xiaoda Yang, Quanyu Dai, Zhenhua Dong, Zhimeng Zhang, and Zhou Zhao. 2025. EAGER-LLM: Enhancing Large Language Models as Recommenders through Exogenous Behavior-Semantic Integration. In *Proceedings of the ACM on Web Conference 2025* (Sydney NSW, Australia) *(WWW '25)*. Association for Computing Machinery, New York, NY, USA, 2754–2762. doi:10.1145/3696410.3714933

[11] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

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

[13] Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do llms understand user preferences? evaluating llms on user rating prediction. *arXiv preprint arXiv:2305.06474* (2023).

[14] Jieyong Kim, Hyunseo Kim, Hyunjin Cho, SeongKu Kang, Buru Chang, Jinyoung Yeo, and Dongha Lee. 2025. driven Personalized Preference Reasoning with Large Language Models for Recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1697–1706.

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

[16] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2024. How Can Recommender Systems Benefit from Large Language Models: A Survey. *ACM Trans. Inf. Syst.* (July 2024). doi:10.1145/3678004 Just Accepted.

[17] Yiyu Liu, Qian Liu, Yu Tian, Changping Wang, Yanan Niu, Yang Song, and Chenliang Li. 2021. Concept-aware denoising graph neural network for micro-video recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 1099–1108.

[18] 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*, Vol. 39. 12246–12254.

[19] Zhongzhou Liu, Hao Zhang, Kuicai Dong, and Yuan Fang. 2024. Collaborative Cross-modal Fusion with Large Language Model for Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1565–1574.

[20] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

[21] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[22] Zihan Wang, Jinghao Lin, Xiaocui Yang, Yongkang Liu, Shi Feng, Daling Wang, and Yifei Zhang. 2025. Enhancing LLM-based Recommendation through Semantic-Aligned Collaborative Knowledge. *arXiv preprint arXiv:2504.10107* (2025).

[23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1800, 14 pages.

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

[25] Xiaohan Yu, Li Zhang, and Chong Chen. 2025. Explainable ctr prediction via llm reasoning. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*. 707–716.

[26] Yang Zhang, Keqin Bao, Ming Yan, Wenjie Wang, Fuli Feng, and Xiangnan He. 2024. 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)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 9181–9191. doi:10.18653/v1/2024.acl-long.497

[27] 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* 37, 5 (2025), 2329–2340. doi:10.1109/TKDE.2025.3540912

[28] Zizhuo Zhang and Bang Wang. 2023. Prompt learning for news recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 227–237.

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