UCGRec: User-Centric Graph Learning for LLM-based Sequential Recommendation

Anonymous ACL submission

Abstract

Recently, Large Language Models (LLM) have emerged as a promising paradigm for sequential recommendation. In sequential recommendation, effectively integrating diverse user preferences is essential for improving LLM performance, as users often exhibit multiple interests across different contexts. However, most existing LLM-based methods rely primarily on item descriptions or utilize user preferences independently. As a result, they overlook the relationships among preferences and fail to filter out less-relevant items that introduce noise. This makes it difficult to accurately capture the user's interests, leading to suboptimal recommendations. To overcome these limitations, we propose **UCGRec** (**U**ser-**C**entric **G**raph Learning for LLM-based Sequential **Rec**ommendation), a novel method that effectively integrates diverse user-relevant preference signals into a unified user-centric graph. Then, we inject the graph-based knowledge into the LLM through end-to-end training with graph neural networks. We conduct extensive experiments on four widely used sequential real-world recommendation datasets. Our experimental results demonstrate that UCGRec significantly outperforms conventional and state-of-the-art LLM-based methods.

1 Introduction

LLM-based recommendation has recently attracted considerable attention (Dai et al., 2023; Hou et al., 2024; Liu et al., 2023; Bao et al., 2023; Geng et al., 2022; Harte et al., 2023), because the knowledge of pre-trained LLMs can enable more sophisticated and personalized recommendations. Early studies relied primarily on in-context learning (Dai et al., 2023; Hou et al., 2024; Liu et al., 2023) and fine-tuning (Bao et al., 2023; Geng et al., 2022; Harte et al., 2023), using item-level textual content such as titles and descriptions.

Since user preference is critical in recommender systems, more recent work has sought to capture diverse preferences—including long-term, short-term, and collaborative signals—from user histories and supply them to the LLM with item content. Some studies tried to capture long-term and short-term preference to provide users' temporal dynamics (Zheng et al., 2024; Chu et al., 2023; Liu et al., 2025). By extracting long-term and short-term preferences from user sequences, they enabled the model to better understand user preferences. Other studies incorporated pre-trained item embeddings obtained from conventional sequential recommenders (Kang and McAuley, 2018; Hidasi, 2015) for utilizing collaborative preferences (Liao et al., 2024; Kim et al., 2024, 2025). Since conventional methods learn item embeddings from all user-item interactions, these embeddings may inherently reflect collaborative preferences. By integrating such item embeddings in the prompt, LLMs can indirectly leverage collaborative preferences.

Although LLMs possess extensive general knowledge, they have some limitations in understanding user preference signals because they have not been trained to effectively integrate and leverage user preference signals. So, we need to consider the following key issues when providing user information to an LLM:

1. Filtering irrelevant interactions. Simply feeding every item interaction of a user to the LLM can impede its ability to infer user preferences. For instance, a user's history can include noisy items clicked accidentally or explored out of interest (Ye et al., 2023; Zhang et al., 2023, 2021). These less-relevant items increase data complexity, making it harder for the model to focus on meaningful signals. However, previous approaches (Zheng et al., 2024; Chu et al., 2023; Liu et al., 2025) often fed complete interaction sequences to the LLM, overlooking this limitation and potentially leading to suboptimal performance. The problem is even more critical for collaborative information. All user information is unlikely to benefit the target user (Ye et al., 2023; Zhang et al., 2023, 2021), and providing every user interaction to the LLM is infeasible. Therefore, we need methods to distill and provide those signals that are highly relevant to the target user to help the LLM better understand the user.

2. Providing diverse user preferences. User preferences are the key factors of sequential recommendation, as they enable a better understanding of user behavior and improve personalized recommendations. Long-term preferences capture consistency in user interests, and short-term preferences reflect more recent interests. Collaborative preferences enrich the target user by incorporating information from similar users. To fully leverage the benefits of each user preference, it is crucial to effectively capture diverse user preferences and the relationships among them. Despite the availability of diverse user preference signal, prior LLM based studies utilized only a single type of user preference (Li et al., 2023; Liu et al., 2025; Zheng et al., 2024; Chu et al., 2023; Liao et al., 2024; Kim et al., 2024, 2025), and failed to capture various user preference and their relationships.

3. Maximizing LLM interpretability. For more personalized recommendations, it is necessary to provide the LLM with diverse preference signals. However, simply listing each preference separately or providing them indirectly via item embeddings does little to help the LLM integrate and reason over the combined information. Besides, if we list too much data, it will increase the burden of the LLM to process the input tokens. It also makes it difficult to understand the relations among diverse signals and limits the possible performance gain. Thus, diverse user-preference information should be integrated in a holistic manner for the LLM to fully utilize them.

To address these issues, we propose a novel LLM-based recommendation method, **UCGRec** (**U**ser-**C**entric **G**raph Learning for LLM-based Sequential **Rec**ommendation) to effectively provide rich information to LLMs and enhance personalized sequential recommendation performance. To filter out irrelevant interactions, we extract diverse user preferences highly relevant to the target user by leveraging similarity-based temporal preferences and interaction-weighted collaborative preferences. Then, we integrate these preferences into a unified graph that enables the LLMs to effectively understand diverse user preferences (He et al., 2024; Tian et al., 2024; Liu et al., 2024). To the best of our knowledge, our study is the first LLM-based sequential recommendation approach that integrates both user preference and collaborative information. To demonstrate the superiority of our proposed method, we conduct experiments on four widely used real-world datasets in sequential recommendation: MovieLens, Steam, LastFM, and Amazon Toys. The results demonstrate that UCGRec outperforms not only conventional methods but also recent state-of-the-art LLM-based sequential recommendation methods.

2 Related Work

Most prior approaches utilized LLMs through in-context learning (Dai et al., 2023; Hou et al., 2024; Liu et al., 2023), or fine-tuned LLMs for recommendation tasks (Geng et al., 2022; Harte et al., 2023; Bao et al., 2023). However, these approaches focus solely on item information and therefore overlook diverse user preferences and often fail to accurately capture user interests. Recent studies have proposed LLM-based sequential recommendation with user preferences to improve their understanding of user behavior (Chu et al., 2023; Zheng et al., 2024; Liu et al., 2025; Liao et al., 2024; Kim et al., 2024, 2025). Chu et al. (2023) proposed the Tempura model to capture long-term preferences by employing a global interest demonstration, which randomly samples user historical behaviors. Zheng et al. (2024) leveraged text-rich information, such as item titles and descriptions, to represent user preferences, and captured long-term and short-term preferences. Liu et al. (2025) summarize users' interaction sequences to extract long-term preferences. This approach provides long-term and short-term preferences from user sequences, they enabled the model to better understand user preferences. Liao et al. (2024) proposed a hybrid method combining ID-based embeddings with textual features to integrate collaborative signals and LLM knowledge. Kim et al. (2024) leveraged collaborative information from traditional models without fine-tuning the LLMs. Kim et al. (2025) integrates sequential information into LLMs by distilling the user representations extracted from a pre-trained item ID embeddings into LLMs. However, these methods derive preferences from all items and transitions, introducing less relevant items and increasing data complexity. Furthermore, they rely on a single type of preferences. As a result, they interrupt the ability of LLMs to effectively capture user preferences and remain suboptimal.

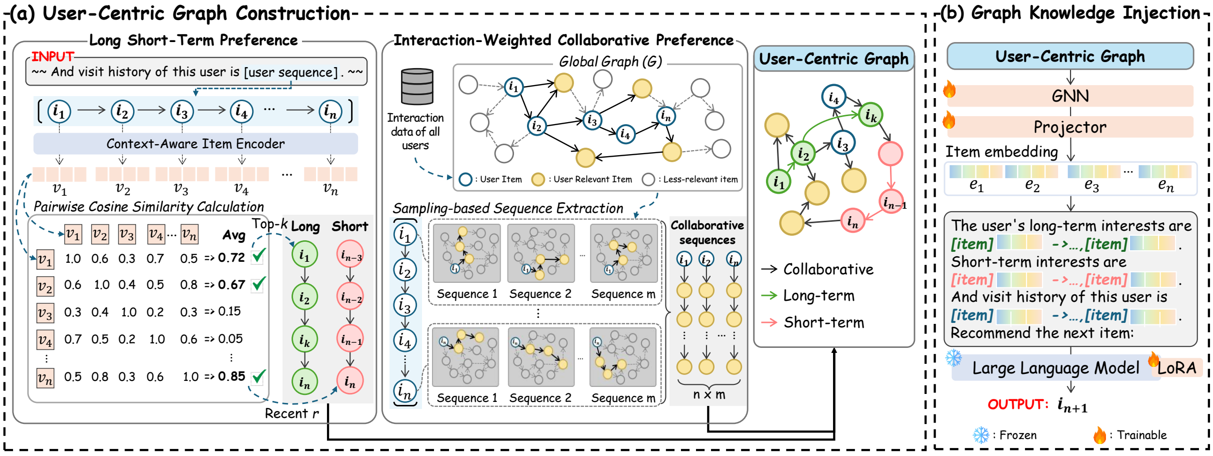


Figure 1: **Overview of UCGRec framework.** (a) In User-Centric Graph Construction, we extract the long short-term preference and the interaction-weighted collaborative preference from a user sequence. These preferences are unified into a user-centric graph (Sec. 3.1). (b) Graph Knowledge Injection jointly trains GNN, projector layer, and LoRA adapter of LLM to inject the graph knowledge into LLM (Sec. 3.2).

3 Proposed Method

Task formulation. Let U and I represent the set of users and items, respectively. For a given user $u \in U$, the user's sequence (interaction history) is denoted as $S^u = [i_1, i_2, \dots, i_n]$, where the item $i \in I$. Our task is to predict the next item, i_{n+1}.

3.1 User-Centric Graph Construction

Figure 1 shows overview of UCGRec framework. To construct the user-centric graph, we first extract both long-term and short-term interest patterns to reflect the temporal dynamics of personal user preferences. Long-term preference contains information about consistent interests that remain stable over time, while short-term preference reflects recent interests that continuously shift. We also extract collaborative item relationships to incorporate structural information from similar user behaviors. Finally, the extracted preference information is unified to construct a user-centric graph.

Long short-term preference. For extracting the long-term sequence, the content information of the items should be considered to select items within the sequence that reflect consistent interests that characterize long-term preferences. As the item token (item's title) alone lacks sufficient content information, we leverage LLMs to generate comprehensive item descriptions that include detailed content information, such as genres and summaries. Then, we select items with consistent interests rather than irrelated items by computing item similarity based on generated content information. The prompt for these item descriptions is detailed in Appendix A.1.

To compute similarity based on item content information, we obtain the textual representation of the generated descriptions using a context-aware item encoder, which is a pre-trained model, as shown in the equation:

$$v_k = \text{Encoder}(\text{LLM}(i_k)),\ i_k \in S^u \tag{1}$$

where v_k denotes the content-aware item embedding for item i_k; S^u is the sequence of the target user u; and $\text{LLM}(i_k)$ is the output text of LLM (e.g., detailed item descriptions) when the title of item i_k is given as input. Through this process, we obtain content-aware item embeddings v_k for all items in the user sequence S^u, using only the item title without additional metadata. Next, we compute the pairwise cosine similarity between the textual representations v of each item using $\text{sim}(\cdot)$. Then, by computing the average similarity among each item, we identify items that share consistently similar semantics within the sequence.

$$\mu_{\text{sim}}(v_k) = \frac{1}{n} \sum_{j=1}^{n} \text{sim}(v_k, v_j) \tag{2}$$

Then, we order the top-k most similar items based on the similarity score μ to construct the long-term sequence S^u_{long}.

For extracting short-term sequence preferences, it is crucial to focus on the user's recent interests, which are influenced by their recently interacted items. We design a simple yet effective approach by extracting the most recent r items from the user sequence as the short-term sequence. We construct a short-term sequence S^u_{short} consisting of items that reflect the user's recent interests.

$$S^u_{\text{short}} = [i_{n-r+1}, \cdots, i_{n-1}, i_n] \tag{3}$$

Interaction-weighted collaborative preference. To effectively capture the collaborative preferences among meaningful neighbor nodes with stronger interactions, we propose a weight sampling-based collaborative sequence extraction method. Incorporating all collaborative information from other users can be inefficient, as it may introduces noisy or less-relevant item. This complexity interrupts the LLM's ability to accurately capture target user preferences, potentially degrading the recommendation performance (He et al., 2024; Tian et al., 2024; Liu et al., 2024). To effectively leverage information from users with similar behavior, we generate a collaborative sequence guided by the target user's interaction history.

First, to utilize the interaction data of all user sequences, a global graph G is pre-constructed to structurally represent all item interactions. The global graph is a weighted directed graph where items in I are the nodes. We add a directed edge between i_j and i_k if $[i_j, i_k]$ is a subsequence of $S^u, u \in U$. The weight of each edge is determined by its frequency across all user sequences in the dataset.

A graph walk is initiated from each item node in the user sequence S^u on the global graph G. Starting from item i_k within S^u, a sequence of length z is constructed to generate a collaborative sequence S^u_{col}. This process is repeated m times for each item in S^u. To reflect not only interaction-weighted relationships between items but also diverse items, we allow the next node selection during the walking process. With probability α, we select either a random neighbor or the next item with the highest edges weight. To mitigate the bias toward specific users, we set $\alpha = 0.6$, which encourages item diversity.

$$P(i_{t+1} \mid i_t) = \begin{cases} \alpha & i_{t+1} = \arg\max_{j \in N(i_t)} w(i_t, j) \\ 1 - \alpha & i_{t+1} = \text{rand}(N(i_t)) \end{cases} \tag{4}$$

where $N(i_t)$ represents the set of neighboring nodes of node i_t, $w(i_t, j)$ denotes the number of edges between i_t and its neighboring node j. Then, we extract meaningful collaborative sequences S^u_{col} by exploring multi-step paths guided by weight sampling.

User-centric graph. Although temporal and collaborative preferences are both highly relevant to the target user, modeling them separately limits the LLM's ability to fully capture their preferences. Inspired by (He et al., 2024; Tian et al., 2024; Liu et al., 2024), we propose a user-centric graph that effectively integrates two types of preferences, thereby enabling LLMs to better utilize this information. First, we represent the item tokens in S^u as nodes and connect the items in interaction order with edges. We merge S^u_{long} and S^u_{short} to reflect dynamic temporal preferences. Then, we integrate all sequences in S^u_{col}, which consist of diverse relevant items, as irrelated items have been filtered out. As the first item of each sequence in S^u_{col} is derived from the target sequence S^u, every first item in S^u_{col} corresponds to an item in S^u. Therefore, all sequences in S^u_{col} can be linked to the nodes in S^u. If an interaction between items overlaps, we reinforce this connection by increasing the edge weight to reflect stronger relationships. Finally, we construct the user-centric graph, which enables the LLMs to better utilize the structural relationships between the user's long short-term and collaborative preferences.

3.2 Graph Knowledge Injection to LLMs

Our user-centric graph structurally represents rich information. To effectively inject this structural knowledge into LLMs, we simultaneously train the GNN, a projector layer, and LoRA (Hu et al., 2021) adapter of LLMs. We first initialize all item nodes in the user-centric graph using pre-trained item embeddings (Kang and McAuley, 2018). The GNN updates these item embeddings by aggregating information from neighboring item nodes based on the graph structure, considering edge weights to capture the relationships between items. To bridge the representation gap between GNN and LLM, we train a projector layer, a trainable MLP, to linearly transform item embeddings into the token space of the LLM. Then, we obtain item embeddings effectively represented from the user-centric graph. The embeddings for the long-term and short-term sequences and the user history sequence are inserted into the prompt for the LLM's input. We concatenate the corresponding item embeddings after the item token of each sequence and incorporate them into the LLM's input prompt. The detailed prompt is illustrated in Appendix A.2. Finally, we jointly train the GNN, Projector, and LLM in an end-to-end manner.

Methods	LastFM		MovieLens		Steam		Toys		Average	
	VR	HR@1	VR	HR@1	VR	HR@1	VR	HR@1	VR	HR@1
Traditional methods										
GRU4Rec	1.0000	0.2616	1.0000	0.3750	1.0000	0.4168	1.0000	0.3387	1.0000	0.3480
SASRec	1.0000	0.2233	1.0000	0.3444	1.0000	0.4010	1.0000	0.4157	1.0000	0.3457
GCE-GNN	1.0000	0.3181	1.0000	0.4295	1.0000	0.4484	1.0000	0.2405	1.0000	0.3591
LLM-based methods										
LLaMA 2-7B	0.3443	0.0246	0.4421	0.0421	0.1653	0.0135	0.0212	0.1582	0.2432	0.0596
GPT-4o	1.0000	0.3443	0.9789	0.2151	0.9671	0.3653	0.8290	0.4127	0.9438	0.3344
TALLRec	0.9825	0.4974	0.9823	0.4695	0.9625	0.4356	0.9879	0.4843	0.9788	0.4717
LLaRA	0.9914	0.4803	0.9766	0.4758	0.9896	0.4886	0.9724	0.5133	0.9825	0.4895
A-LLMRec	0.9883	0.2557	0.9728	0.3790	0.9711	0.4194	0.9725	0.2931	0.9762	0.3368
LLM-SRec	0.9921	0.3590	0.9836	0.4211	0.9942	0.4862	0.9892	0.3749	0.9897	0.4103
UCGRec (ours)	0.9873	**0.5447**	0.9827	**0.4949**	0.9865	**0.5040**	0.9734	**0.5574**	0.9825	**0.5252**

Table 1: Comparison of our proposed method with three traditional and five LLM-based methods on the four benchmark datasets. We evaluate ValidRatio (VR) and HitRatio@1 (HR@1). The best performance is highlighted in **boldface**, and the second-best is marked as underlined. All improvements are statistically significant ($p < 0.05$).

4 Experimental Setup

4.1 Datasets

We conduct experiments on four real-world sequential recommendation datasets to evaluate our method. MovieLens (Harper and Konstan, 2015) is a public dataset for movie recommendation, which contains movie titles based on users' viewing history. Steam (Kang and McAuley, 2018) is a video game dataset collected from the Steam platform, which includes user reviews, ratings, and game titles. LastFM (Cantador et al., 2011) is a music dataset from the Last.fm streaming service that includes user-artist listening interactions and artist names. Toys[1] (He and McAuley, 2016) is a dataset from Amazon's Toys and Games category, containing user reviews, ratings, and product descriptions. Detailed statistics of the datasets are provided in the Appendix A.3.

4.2 Baselines

We compare our proposed method with nine baselines: (1) three traditional sequential recommendation methods—SASRec (Kang and McAuley, 2018), GRU4Rec (Hidasi, 2015), GCE-GNN (Wang et al., 2020); and (2) six LLM-based sequential recommendation methods—LLaMA 2-7B (Touvron et al., 2023), GPT-4o (Achiam et al., 2023), TALLRec (Bao et al., 2023), LLaRA (Liao et al., 2024), A-LLMRec (Kim et al., 2024), LLM-SRec (Kim et al., 2025). LLaMA 2-7B and GPT-4o performed zero-shot inference. TALLRec, LLaRA, A-LLMRec, and LLM-SRec adopt fine-tuning strategies. TALLRec and LLaRA fine-tune the LLM with LoRA, while A-LLMRec and LLM-SRec keep the LLM frozen and trains only the encoder and projector. Detailed explanations of each method are provided in the Appendix A.4.

[1]The Amazon Toys dataset we used is based on the 2018 Amazon Review dataset `https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/`

4.3 Evaluation

Metrics. To evaluate the recommendation performance, we used two widely adopted metrics in LLM-based sequential recommendation, HitRatio@1 (HR@1) and ValidRatio (VR). HitRatio@1 measures whether the top-1 recommended item matches the user's ground-truth item. ValidRatio evaluates the proportion of valid responses where the predicted item appears within the candidate set. A ValidRatio of 0.95 or higher indicates that performance metrics such as HitRatio@1 can be trusted. Since LLM-based methods generate a single candidate item through appropriate prompting, HitRatio@1 is adopted as the evaluation metric.

Implementation Details. To mitigate the impact of randomness and non-determinism in LLMs, we set the temperature to 0 and report the average results over five runs with different random seeds. More details are provided in Appendix A.5.

5 Experiment Results

5.1 Overall results

Table 1 presents the overall results of our proposed UCGRec against the latest state-of-the-art methods on four real-world datasets. Notably, UCGRec achieves the highest HR@1 on every individual

Components			Datasets				
L	*S*	*C*	LastFM	MovieLens	Steam	Toys	Average
✓	✗	✗	0.5146	0.4750	0.4962	0.5121	0.4995
✗	✓	✗	0.5328	0.4215	0.4829	0.5497	0.4967
✗	✗	✓	0.5344	0.4586	0.4675	0.5504	0.5027
✓	✓	✓	**0.5447**	**0.4949**	**0.5040**	**0.5574**	**0.5252**

Table 2: Ablation study on temporal and collaborative patterns of the user (HitRatio@1). *L*, *S*, and *C* indicate long-term, short-term, and collaborative preference, respectively. "✓" indicates that the component is included in UCGRec, while "✗" indicates that it is not.

dataset as well as in the overall average, highlighting its consistently superior performance. Additionally, UCGRec also achieves a reasonable ValidRatio ranging from 0.96 to 0.99 across all datasets.

Our method, UCGRec, achieves a significant performance improvement compared to traditional methods, such as GRU4Rec, SASRec, and GCE-GNN. These results indicate that traditional methods, which rely solely on user sequence information, fail to effectively utilize rich contextual information about items and instead primarily focus on data matching. Compared to LLM-based methods, UCGRec also shows significantly higher average performance scores than the LLaMA 2-7B and GPT-4o, which used zero-shot inference for recommendation tasks. This indicates that general LLMs, without knowledge of sequential recommendation tasks, are unable to perform effectively in recommendation tasks. Also, UCGRec outperforms TALLRec, which fine-tunes LLMs for sequential recommendation tasks by utilizing item description, with a 11.3% higher average performance. These results indicate that simply inserting a user sequence into a textual prompt may prevent LLMs from fully utilizing their general knowledge and reasoning abilities. UCGRec outperforms LLaRA, which utilize collaborative information through item embeddings, achieving a 7.3% higher average performance. These improvements are notable, as HR@1 is a strict metric that evaluates only the top-1 item. This demonstrates that indirectly using collaborative information and overlooking temporal preferences limits the performance of personalized recommendation. A-LLMRec and LLM-SRec also show lower performance, as they trains only task-specific modules instead of the LLMs themselves, and incorporate less relevant collaborative information.

5.2 Ablation Study

We conduct an ablation study to validate the effectiveness of our proposed components. Within our framework, we compare HR@1 results based on different combinations: using all three components (*L* - long-term , *S* - short-term, and *C* - collaborative information), and each individually.

As shown in Table 2, utilizing all components achieves the best performance compared to individually using each component. This indicates that our method fully leverages the strengths of each user preference. Our method including only *L* component shows performance improvement, particularly in the MovieLens and Steam dataset. Since people's movie and game genres tend to remain consistent, allowing *L* better captures long-term preferences on MovieLens and Steam.

On the other hand, in the LastFM and Toys dataset, our method including only *S* component is more effective than that of using only *L*. It is because people's music and toy preferences are influenced by recent popularity.

When using only *C* component, our method shows a high overall performance but performs less effectively in the Steam dataset. Since Steam has many interactions but relatively few items, it results in the construction of a highly sparse graph. This highlights the importance of effectively integrating all components, as their interaction can yield greater benefits than utilizing each one individually. Our proposed method, which leverages all three components, demonstrates significantly higher performance compared to when they are used independently.

6 Further Analysis

In this section, we conduct in-depth analyses of UCGRec. We validate our long-term preference using its variants (Sec. 6.1) and evaluate our interaction-weighted collaborative preference against one derived from a global graph that includes all user-item interactions (Sec. 6.2). We further investigate our user-centric graph (Sec. 6.3) and evaluate UCGRec under cold-start settings to demonstrate effectiveness in real-world scenarios (Sec. 6.4). For more robust evaluation, we use Hit@K for accuracy and NDCG@K for ranking quality (Sec. 6.5). Additional results and case studies are provided in Appendix (A.6-A.10).

Method	LastFM	MovieLens	Steam	Toys
(L_{rand})	0.4882	0.4289	0.4861	0.4450
(L_{all})	0.4795	0.4484	0.4880	0.4802
(L_{ours})	0.5146	0.4750	0.4962	0.5121
($L_{\text{rand}} + S + C$)	0.4869	0.4745	0.5008	0.5493
($L_{\text{all}} + S + C$)	0.5146	0.4743	0.4984	0.5304
($L_{\text{ours}} + S + C$)	**0.5447**	**0.4949**	**0.5040**	**0.5574**

Table 3: Effectiveness of our context-aware long-term extraction method (HitRatio@1). (L_{our}) represents using only our long-term preference in UCGRec. (L_{rand}) uses randomly sampled long-term preferences. (L_{all}) uses all items as long-term preferences.

Method	LastFM	MovieLens	Steam	Toys
LLaRA	0.4803	0.4758	0.4886	0.5133
A-LLMRec	0.2557	0.3790	0.4194	0.2931
LLM-SRec	0.3590	0.4211	0.4862	0.3749
UCGRec (C_{global})	0.4391	0.4765	0.4897	0.5179
UCGRec (C_{ours})	**0.5447**	**0.4949**	**0.5046**	**0.5574**

Table 4: Performance of collaborative information from different types. Baselines indirectly utilizes sequential collaborative information from pre-trained models. (C_{global}) obtains collaborative information from global graph that includes all user-item interactions.

Dataset	Scale	UCGRec (C_{global}) HR@1	UCGRec (C_{global}) time	UCGRec (C_{ours}) HR@1	UCGRec (C_{ours}) time
LastFM	70%	0.3826	0.6366	0.4833	0.0557
	30%	0.3781	0.3365	0.4262	0.0346
MovieLens	70%	0.3764	0.6029	0.4230	0.0784
	30%	0.3806	0.2973	0.4103	0.0465

Table 5: Performance comparison under increasing user scenarios (HitRatio@1) and graph construction time (seconds per user).

6.1 Effectiveness of Long-term Preference

To assess the effectiveness of our context–aware long-term preference extraction, we compare variants of long-term preference in our framework. When using the long-term preference extracted by our method L_{our} versus those obtained via the random sampling approach L_{rand} used in Tempura (Chu et al., 2023). L_{all} is using all items as long-term preferences. ($L + S + C$) represent the variants of long-term preferences integrated with short-term preference and collaborative preference in our framework. As shown in Table 3, L_{our} consistently outperforms L_{rand} and L_{all}, demonstrating that our method selects highly relevant items and better captures long-term preferences. Specifically, L_{our} outperforms L_{all}, indicating that not all items are helpful in capturing user preferences. When integrated with short-term and collaborative information, ($L_{\text{our}} + S + C$) outperforms all other variants. Although integrating L_{all} or L_{rand} with short-term and collaborative information generally improves performance, L_{our} achieves the best results. This indicates that our long-term sequences facilitate meaningful connections within the unified graph. Analysis of long short-term sequences length are reported in Appendix A.6.

6.2 Effectiveness of Collaborative Preference

To evaluate how effectively our interaction-weighted collaborative preference, we compare our proposed user-centric graph with a global graph. UCGRec (C_{our}) uses a user-centric graph that incorporates long-term, short-term, and interaction-weighted collaborative preferences. On the other hand, UCGRec (C_{global}) uses a global graph instead of the user-centric graph within our framework.

As shown in Table 4, UCGRec notably outperforms the global graph, demonstrating that focusing on target-relevant user interactions leads to more personalized and effective representation learning. Additionally, UCGRec surpasses the baselines, sequential collaborative information, suggesting that incorporating all users' sequences without structural information may dilute target user preferences. These results highlight the importance of selectively leveraging collaborative information through a user-centric graph-based approach. Analysis of collaborative preference lengths are provided in Appendix A.7.

6.3 Analysis of User-Centric Graph

Scalability. To demonstrate the scalability and efficiency of our proposed graph, we conduct additional experiments on different dataset size. To simulate a scenario with an increasing number of users, we conduct experiments setting the data to 30% and 70% of the full dataset. As shown in Table 5, UCGRec (C_{ours}) consistently outperforms UCGRec (C_{global}) as the dataset size increases. This demonstrates that UCGRec captures highly relevant user preference information. The result indicates that it is not only scalable but also capable of achieving accurate recommendations in large-scale datasets.

Effectiveness. To verify whether the structural information from our proposed user-centric graph is effectively utilized by the LLM, we analyze the attention scores between the predicted token and the input tokens. The results of analysis are provided in Appendix A.8.

Method	MovieLens		Toys		Steam	
	Cold	Warm	Cold	Warm	Cold	Warm
TALLRec	0.1935	**0.4732**	0.3909	0.4573	0.1641	0.4347
A-LLMRec	0.1375	0.1056	0.1980	0.2481	0.1541	0.4228
LLaRA	0.2225	0.4399	0.4295	0.5009	0.2005	0.4787
LLM-SRec	0.2308	0.4149	0.2466	0.3612	**0.2800**	0.4825
UCGRec	**0.2735**	0.4502	**0.4717**	**0.5495**	0.2151	**0.4900**

Table 6: Comparison of performance under cold and warm item scenarios (HitRatio@1). Cold and warm items are labeled based on interaction frequency, with warm items belonging to the top 35% and cold items to the bottom 35%.

Time-efficiency. Furthermore, the time for user-centric graph construction remains nearly constant as the data size increases, and the inference time is 2.34 seconds per users (Appendix A.9). This indicates that our method is highly time-efficient and practical in increasingly large-scale scenarios.

6.4 Performance in Cold-start Problem

Cold/Warm Item Scenario. We evaluate UCGRec under cold and warm item scenarios. Following the cold-start setup in A-LLMRec (Kim et al., 2024), we label items as warm (top 35%) or cold (bottom 35%) according to item interaction frequency. As shown in Table 6, UCGRec consistently outperforms all baselines in both scenarios. UCGRec achieves strong performance even for cold items with insufficient information by combining diverse user-centric preferences and the LLM's world knowledge. This indicates that the proposed user-centric graph effectively captures diverse user preferences, enabling the LLM to better interpret user interests under the cold/warm item scenarios.

Cold User Scenario. To comprehensively evaluate the cold-start problem, we additionally conduct experiments under the cold-user scenario. To simulate the cold user scenario, we follow the same setup as used in A-LLMRec (Kim et al., 2024) . We select users with three interactions, use the first two as input, and the last (most recent) item as the ground truth. We evaluate on cold users with disjoint user splits to prevent information leakage. In cold-user scenarios, as shown in Table 7, UCGRec outperforms all baselines in LastFM and Toys. Unlike previous LLM-based methods that rely solely on textual or collaborative information, our method leverages diverse user-centric preferences. This enables reliable performance even with extremely limited interactions.

Method	LastFM	MovieLens	Toys	Steam
TALLRec	0.4353	0.3951	0.3407	0.3218
A-LLMRec	0.0428	0.3516	0.1993	0.3285
LLaRA	0.4686	**0.4593**	0.3353	0.4042
LLM-SRec	0.3133	0.2990	0.2655	**0.4398**
UCGRec	**0.4853**	0.4390	**0.4059**	0.3780

Table 7: Comparison of performance under cold user scenarios (HitRatio@1). A cold user is defined as a user who has interacted with exactly three items.

Dataset	Method	Hit@K		NDCG@K	
		K=10	K=20	K=10	K=20
LastFM	TALLRec	0.4393	0.5999	0.2331	0.2742
	LLaRA	0.5164	0.6770	0.3026	0.3434
	LLM-SRec	0.4809	0.6999	0.3049	0.3850
	UCGRec	**0.6230**	**0.7656**	**0.3751**	**0.4112**
MovieLens	TALLRec	0.2421	0.3411	0.1162	0.1416
	LLaRA	0.4126	0.5663	0.2076	0.2468
	LLM-SRec	0.6400	0.7747	0.3894	0.4233
	UCGRec	**0.6653**	**0.8084**	**0.3988**	**0.4346**

Table 8: Performance with Hit@K and NDCG@K (K=10,20). Bold indicates the best performance.

6.5 Evaluation with Additional Metrics

For more robust evaluation, we adopt Hit@K to measure recommendation accuracy and NDCG@K to account for ranking quality by giving higher weight to top-ranked items. Since LLM-based methods generate a single candidate item, they are inherently optimized for top-1 prediction, making it difficult to evaluate ranking quality. To address this, following LLM-SRec (Kim et al., 2025), we add a projection layer that maps the LLM output to the item embedding space, enabling similarity-based top-K retrieval from the entire item pool. As shown in Table 8, UCGRec outperforms baselines in both metrics. This suggests that our approach captures user preferences more accurately, leading to better top-K retrieval performance.

7 Conclusion

We proposed UCGRec to address key challenges in LLM-based recommendation: filtering irrelevant interactions, providing diverse user preferences, and enhancing interpretability for the language model. UCGRec integrates these preferences into a unified user-centric graph that captures structural relationships between long/short-term and collaborative preferences, enabling effective user behavior interpretation. Extensive experiments on four real-world datasets demonstrated its superiority over state-of-the-art baselines.

Limitations

While our proposed method achieves strong performance, several limitations remain inherent to LLM-based sequential recommendation: (1) One limitation is its reliance on LLMs, which requires more parameters and training time compared to traditional models trained from scratch. Nevertheless, this cost is justified by the improved recommendation performance achieved through leveraging their pre-trained knowledge. (2) Although recommendation datasets can include various type of metadata such as images and category information, our current approach only utilizes textual information. In future work, we plan to extend our method to multi-modal sequential recommendation by incorporating additional modalities.

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.

Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). In *Proceedings of the fifth ACM conference on Recommender systems*, pages 387–388.

Zhendong Chu, Zichao Wang, Ruiyi Zhang, Yangfeng Ji, Hongning Wang, and Tong Sun. 2023. Improve temporal awareness of llms for domain-general sequential recommendation. In *ICML 2024 Workshop on In-Context Learning*.

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*, pages 1126–1132.

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

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

Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1096–1102.

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*, pages 507–517.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

B Hidasi. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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

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

Sein Kim, Hongseok Kang, Kibum Kim, Jiwan Kim, Donghyun Kim, Minchul Yang, Kwangjin Oh, Julian McAuley, and Chanyoung Park. 2025. Lost in sequence: Do large language models understand sequential recommendation? *arXiv preprint arXiv:2502.13909*.

Xinhang Li, Chong Chen, Xiangyu Zhao, Yong Zhang, and Chunxiao Xing. 2023. E4srec: An elegant effective efficient extensible solution of large language models for sequential recommendation. *arXiv preprint arXiv:2312.02443*.

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

Jiahao Liu, Xueshuo Yan, Dongsheng Li, Guangping Zhang, Hansu Gu, Peng Zhang, Tun Lu, Li Shang, and Ning Gu. 2025. Improving llm-powered recommendations with personalized information. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2560–2565.

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

Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V Chawla. 2024. Can we soft prompt llms for graph learning tasks? In *Companion Proceedings of the ACM Web Conference 2024*, pages 481–484.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V Chawla, and Panpan Xu. 2024. Graph neural prompting with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19080–19088.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 169–178.

Yaowen Ye, Lianghao Xia, and Chao Huang. 2023. Graph masked autoencoder for sequential recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 321–330.

Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 367–377.

Yipeng Zhang, Xin Wang, Hong Chen, and Wenwu Zhu. 2023. Adaptive disentangled transformer for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3434–3445.

Zhi Zheng, Wenshuo Chao, Zhaopeng Qiu, Hengshu Zhu, and Hui Xiong. 2024. Harnessing large language models for text-rich sequential recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3207–3216.

A Appendix

A.1 Item Description Prompt

To accurately extract a user's long-term sequence, it is essential to consider the content information of items. Since the item token (such as the title) does not sufficiently capture the item's content, we utilize LLMs to generate rich item descriptions. As described in Figure 2, by providing the above prompt as input to the LLM, we obtain detailed content information of item as the LLM's output. The prompt for generating these detailed item descriptions, where i_p denotes the p-th user item token.

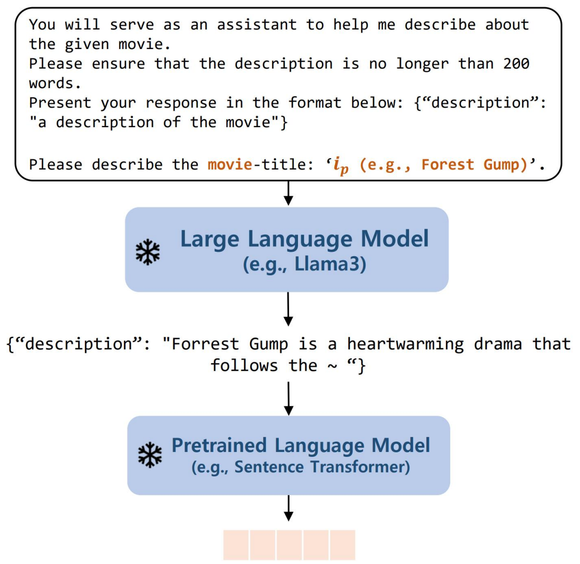


Figure 2: Item description prompt for movielens dataset.

A.2 Prompt for LLM-based Recommendation

Table 12 presents the prompts used in our UCGRec and other LLM-based methods. The prompt for TALLRec and zero-shot inference methods are derived from the LLM-based prompt without the item embeddings $[e_i]$. On the other hands, the prompt with item embeddings $[e_i]$ are integrated into the LLM's token space, allowing the model to understand item embedding $[e_i]$ from sequential-based or graph-based information. Our prompt enhances the LLM's recommendation reasoning ability, enabling it to infer user preferences from their sequences based on long short-term preferences. The designed prompt serves as the input to the LLM, and the expected output is the next interaction item.

Table 9: Statistics of the sequential recommendation datasets used in our experiments.

Datasets	# Users	# Items	# Interactions
MovieLens	943	1,682	100,000
Steam	11,938	3,581	274,726
LastFM	1,220	4,606	73,510
Toys	97,682	45,497	118,940

A.3 Datasets

Table 9 summarizes the statistics of the datasets used in this paper. For training and evaluation, the data is split into train, valid, and test sets in an 8:1:1 ratio. To ensure a fair comparison, we adopt the same pre-processing setup used in LLaRA (Liao et al., 2024). We remove users with fewer than 20 item interactions. Considering the characteristics of sequential recommendation data, we sort user interactions in chronological order to construct user sequences. The sequence is limited to the recent 10 interactions. We apply padding for sequences with fewer than 10 interactions.

A.4 Baselines

The traditional methods adopt full fine-tuning, where both the item embeddings and the model are trained. The LLM-based methods can be categorized into two types based on their training strategies: zero-shot and fine-tuning. LLaMA 2-7B and GPT-4o performed zero-shot inference. In contrast, TALLRec, LLaRA, and A-LLMRec adopt fine-tuning strategies. Among them, TALLRec and LLaRA fine-tune the LLM with LoRA, while A-LLMRec and LLM-SRec keep the LLM frozen and trains only the encoder and projector.

- **SASRec** (Kang and McAuley, 2018) utilizes a self-attention mechanism to model user preferences based on item sequences.
- **GRU4Rec** (Hidasi, 2015) proposes a GRU-based architecture to capture sequential user behavior.
- **GCE-GNN** (Wang et al., 2020) adopts a graph-based framework to model item transitions for sequential recommendation.
- **LLaMA 2-7B** (Touvron et al., 2023) is a open-source LLM designed for general natural language processing tasks.

- **GPT-4o** (Achiam et al., 2023) is a state-of-the-art LLM released by OpenAI for complex language understanding and generation.
- **TALLRec** (Bao et al., 2023) fine-tunes LLMs using instruction tuning to better align them with recommendation objectives.
- **LLaRA** (Liao et al., 2024) integrates collaborative information through a hybrid prompting method that combines pre-trained ID-based item embeddings with textual features.
- **A-LLMRec** (Kim et al., 2024) enables LLMs to leverage collaborative information from ID-based item embeddings without extensive fine-tuning.
- **LLM-SRec** (Kim et al., 2025) integrates sequential information into LLMs by distilling the user representations extracted from a pre-trained CF-SRec model into LLMs.

A.5 Implementation Details

Candidate set for evaluation. Following the setting of LLaRA (Liao et al., 2024) for fair comparison, for each test instance, we construct a candidate set consisting of 20 items: one ground-truth item and 19 randomly sampled non-interacted items. Then, we use the same candidate set across all baseline models, including both traditional sequential recommenders and LLM-based methods.

Pre-trained models. To extract long-term sequences, we used LLaMA 3-8B to generate detailed item descriptions and Sentence-BERT (Thakur et al., 2021) to obtain context-aware representations of these descriptions.

Hyperparameters. The hyperparameters for the lengths of long-term and short-term sequences (k,r) were set as (3,3). For collaborative sequence extraction, we applied weighted sampling with m = 6 iterations and graph walk length z = 10. The next node selection probability α was set to 0.6. For graph knowledge injection, we used 1-hop neighbors for LastFM, and 4-hop neighbors for MovieLens, Steam, and Toys. Item representations were initialized using pre-trained embeddings from SASRec (Kang and McAuley, 2018), with an embedding dimension of 64. For training the LLM-based sequential recommendation model, we used LLaMA 2-7B (Touvron et al., 2023) as the backbone model and applied LoRA (Hu et al., 2021) with a rank of 8. The model was trained using the Adam optimizer (Loshchilov and Hutter) with a learning rate of 3e-4 and trained for up to 8 epochs. The batch size and gradient accumulation steps were set as follows: Toys used (1, 128), MovieLens and LastFM used (8, 16), and Steam used (4, 32), respectively.

A.6 Analysis of Long Short-term Sequences Length

We analyze how performance changes with different values of k and r, the length of long-term and short-term, respectively. Since the maximum length of the target sequence is 10, analyzing all possible combinations of k and r would be highly inefficient. Therefore, we set the minimum and maximum lengths of the long short-term sequences to 3 and 5.

As shown in Table 10, we observe that the optimal hyperparameters for long short-term sequence length depending on the data characteristics. This variability comes from the unique composition of each dataset, where the number of users, items, and their characteristics are differ. Consequently, the ability to capture meaningful interactions changes with the length of the long short-term sequence.

For LastFM, while people tend to maintain consistency in their preferred music genres, songs with shorter listening times allow users to easily explore a variety of music, making it important to consider recent preferences as well. Therefore, LastFM show best performance with k = 3 and r = 3. Similarly, Toys tends to show a preference for items within a specific category, reflecting consistent interests. However, purchases can also be influenced by events such as seasonal trends and new product releases, making it important to consider recent preferences as well. Unlike LastFM, Toys has a significantly larger number of items, requiring a broader selection of items to effectively capture both long short-term preferences. As a result, Toys show best performance with k = 5 and r = 5. MovieLens and Steam share the same k = 5 and r = 3. This indicates that while user preferences are influenced by recent popularity (e.g., movie releases or game launches), both types of items require longer interaction times (e.g., watching a movie or playing a game). As a result, more focus should be on long-term preferences.

Table 10: The analysis of long short-term sequence length. k represents the length of the long-term sequence, r represents the length of the short-term sequence.

k	r	LastFM	MovieLens	Steam	Toys
3	3	**0.6083**	0.5164	0.5233	0.5925
3	5	0.6033	0.5543	0.5072	0.5929
5	5	0.5378	0.5268	0.5169	**0.6009**
5	3	0.5932	**0.5869**	**0.5241**	0.5885

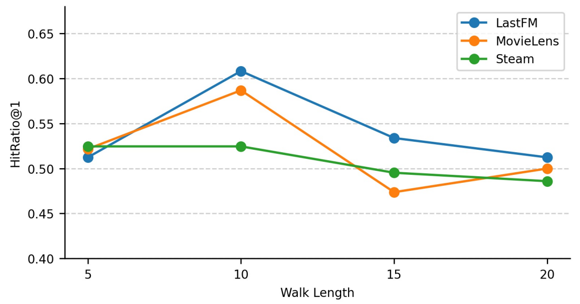


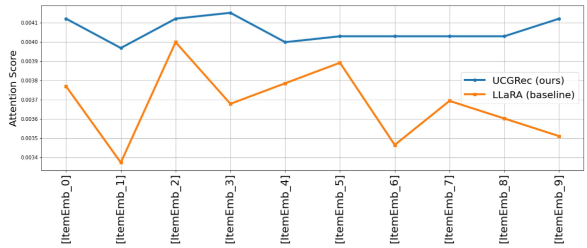
Figure 3: Analysis of walk length. The x-axis represents the length of the walk length, and the y-axis represents the HitRatio@1 score.

A.7 Analysis of Collaborative Preference Lengths

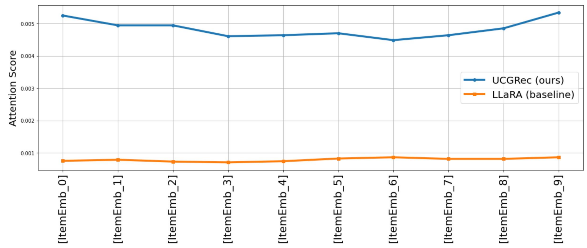
We also analyze how performance changes with different values of the walk step length (z) when extracting collaborative preferences from the global graph on LastFM, MovieLens and Steam dataset. The results of the analysis are presented in Figure 3. These results show that performance increase from 5 to 10, but gradually decreases as the walk length exceeds 10. This suggests that as the walk step length increases, the user-centric unified graph becomes more similar to the global graph, reflecting information from all users not only target-relevant users. Consequently, this dilution of the target user's information and the reflection of unrelated information negatively impact the recommendation performance. In this paper, we appropriately set the graph walk length z to 10 when extracting the collaborative sequence, effectively utilizing target-relevant information within the user-centric graph.

A.8 Impact of Graph Embeddings on LLMs

We calculate the attention scores from the final layer and then examine how much attention the predicted token assigns to each item embedding token [ItemEmb$_n$]. We compare UCGRec and LLaRA, both of which use item embeddings in the input



(a) Attention score - Layer 32 (final), Head 14



(b) Attention score - Layer 32 (final), Head 26

Figure 4: Attention score comparison between UCGRec and LLaRA with respect to [ItemEmb$_n$], where the x-axis indicates item embedding tokens and the y-axis indicates attention scores. The blue line shows the attention scores for [ItemEmb$_n$] from UCGRec (ours), and the orange line shows those from LLaRA.

Table 11: Comparison of per-epoch training time, per-user inference time and trainable parameters on the LastFM dataset. Graph size denotes the number of nodes per user.

	UCGRec (C_{ours})	UCGRec (C_{global})	LLaRA
Graph Size	192.22	4,606	–
Train time (min/epoch)	77.82	104.57	46.47
Inference time (sec/user)	2.34	2.36	2.31
Total Params	6.8 B	6.8 B	6.8 B
Trainable Params	37.1 M	37.1 M	37.0 M
HR@1	0.6083	0.5339	0.5246

prompt. Unlike UCGRec, LLaRA does not incorporate diverse user behavior signals.

As shown in Figure 4, UCGRec exhibits consistently higher attention to [ItemEmb$_n$] across multiple attention heads. In particular, head 26 shows a significant focus on [ItemEmb$_n$]. This highlights that the LLM actively utilizes the structural information from the user-centric graph, enabling it to better capture temporal patterns and collaborative signals for sequential recommendations.

A.9 Cost Efficiency Analysis

Services such as e-commerce and video platforms typically maintain a large number of items, with new items and users continuously being added over time. As a result, recommendation systems need to provide fast and accurate item suggestions, even in large-scale user-item interaction environments. To meet this requirement, both computational efficiency and scalability must be considered. To demonstrate the cost effectiveness of our proposed method, we conduct experiments by varying the graph/model size, and measure the corresponding training/inference time and performance. To verify its effectiveness, we compare our method against (1) UCGRec (C_{global}) which used in Sec 6.2. As shown in Table 11, our UCGRec achieves faster training and inference speeds compared to the global-graph-based UCGRec (C_{global}). This indicates that our user-centric graph includes only target-relevant user interactions, enabling a smaller graph to achieve high performance.

A.10 Case Studies

We select two typical cases to analyze the impact of temporal and collaborative preferences leveraged by large language models (LLMs). For comparison, we examined the generated answer from TALLRec, LLaRA, and UCGRec.

As shown in Table 12, the user sequentially watched *Lassie, Robin Hood: Men in Tights, Sgt. Bilko, Dumb & Dumber, Grease 2, Ace Ventura: Pet Detective, Carrie, The Beverly Hillbillies, Jaws 2, The Brady Bunch Movie*. The user's preferences clearly include the genres of Comedy (e.g., *Robin Hood: Men in Tights, Sgt. Bilko, Dumb & Dumber, Grease 2, Ace Ventura: Pet Detective, The Beverly Hillbillies, The Brady Bunch Movie*) and Horror (e.g., *Carrie, Jaws 2*). Both TALLRec and LLaRA predicted *Phat Beach* and *The Santa Clause* respectively as the next movie to watch, resulting in incorrect recommendations. In contrast, UCGRec successfully recommended *Candyman*. Both TALLRec and LLaRA focused on the Comedy genre, which is the user's most frequently watched genre, recommending *Phat Beach* and *The Santa Clause* as Comedy films. This indicates a failure to consider the user's recent interest, the short-term preference for Horror. However, UCGRec informed the LLM that the user's latest interest was in the horror film *Jaws 2*, enabling the model to recommend the similar horror movie *Candyman*.

Conversely, in Table 13, LLaRA focused on the last watched horror movie (*The Devil's Own*) and recommended the thriller *Shadow Conspiracy*, which led to a incorrect recommendation. UCGRec, on the other hand, focused on the user's long-term preference for Drama and Action genres and recommended *Grosse Pointe Blank*, resulting in a correct recommendation. Moreover, UCGRec leveraged not only temporal preferences but also the viewing histories of users with similar preference, thereby effectively incorporating subtle long and short-term preferences.

These two case studies demonstrate that UCGRec enables the LLM to effectively understand and utilize both temporal and collaborative preferences, leading to more accurate recommendations.

Table 12: An example input prompt for the Movielens dataset used by UCGRec and LLM-based model. Each $[e_n]$ indicates the embedding of item n. Gray embeddings represent simple collaborative information. Blue embeddings represent user-centric graph knowledge that integrates both temporal preferences and collaborative preferences through structural relationships.

LLM-based Prompt w/o embedding (TALLRec)	**Generated Answer**
The visit history of this user is: Lassie, Robin Hood: Men in Tights, Sgt. Bilko, Dumb & Dumber, Grease 2, Ace Ventura: Pet Detective, Carrie, The Beverly Hillbillies, Jaws 2, The Brady Bunch Movie. Please predict the next movie this user will watch. Choose the one answer from the following movie titles: Tigrero: A Film That Was Never Made, Candyman, Cat People, Sunset Park, Hoodlum, Tom & Viv, The Search for One-eye Jimmy, Three Wishes, The Blues Brothers, E.T. the Extra-Terrestrial, National Lampoon's Senior Trip, Speed, Raging Bull, Firestorm, The Santa Clause, For Love or Money, Aladdin and the King of Thieves, Dream Man, Air Bud, Phat Beach. The answer is:	Phat Beach Incorrect
LLM-based Prompt (LLaRA)	**Generated Answer**
The visit history of this user is: Lassie $[e_1]$, Robin Hood: Men in Tights $[e_2]$, Sgt. Bilko $[e_3]$, Dumb & Dumber $[e_4]$, Grease 2 $[e_5]$, Ace Ventura: Pet Detective $[e_6]$, Carrie $[e_7]$, The Beverly Hillbillies $[e_8]$, Jaws 2 $[e_9]$, The Brady Bunch Movie $[e_{10}]$. Please predict the next movie this user will watch. Choose the one answer from the following movie titles: Tigrero: A Film That Was Never Made $[e_{241}]$, Candyman $[e_{47}]$, Cat People $[e_{25}]$, Sunset Park $[e_{94}]$, Hoodlum $[e_{209}]$, Tom & Viv $[e_{733}]$, The Search for One-eye Jimmy $[e_{51}]$, Three Wishes $[e_{26}]$, The Blues Brothers $[e_{219}]$, E.T. the Extra-Terrestrial $[e_{89}]$, National Lampoon's Senior Trip $[e_{18}]$, Speed, Raging Bull $[e_{64}]$, Firestorm $[e_{36}]$, The Santa Clause $[e_{84}]$, For Love or Money $[e_{33}]$, Aladdin and the King of Thieves $[e_{417}]$, Dream Man $[e_{128}]$, Air Bud $[e_{348}]$, Phat Beach $[e_{439}]$. The answer is:	The Santa Clause Incorrect
Our Prompt (UCGRec)	**Generated Answer**
The user's long-term interests are Lassie $[e_1]$, Dumb & Dumber $[e_4]$, Grease 2 $[e_5]$, Ace Ventura: Pet Detective $[e_6]$, The Beverly Hillbillies $[e_8]$, and short-term interests are The Beverly Hillbillies $[e_8]$, Jaws 2 $[e_9]$, The Brady Bunch Movie $[e_{10}]$, and visit history of this user is Lassie $[e_1]$, Robin Hood: Men in Tights $[e_2]$, Sgt. Bilko $[e_3]$, Dumb & Dumber $[e_4]$, Grease 2 $[e_5]$, Ace Ventura: Pet Detective $[e_6]$, Carrie $[e_7]$, The Beverly Hillbillies $[e_8]$, Jaws 2 $[e_9]$, The Brady Bunch Movie $[e_{10}]$. Please predict the next movie this user will watch. Note that both of user long-term interests and short-term interests should be holistically considered for a more comprehensive understanding of user behavior. Choose the one answer from the following movie titles: Tigrero: A Film That Was Never Made $[e_{241}]$, Candyman $[e_{47}]$, Cat People $[e_{25}]$, Sunset Park $[e_{94}]$, Hoodlum $[e_{209}]$, Tom & Viv $[e_{733}]$, The Search for One-eye Jimmy $[e_{51}]$, Three Wishes $[e_{26}]$, The Blues Brothers $[e_{219}]$, E.T. the Extra-Terrestrial $[e_{89}]$, National Lampoon's Senior Trip $[e_{18}]$, Speed, Raging Bull $[e_{64}]$, Firestorm $[e_{36}]$, The Santa Clause $[e_{84}]$, For Love or Money $[e_{33}]$, Aladdin and the King of Thieves $[e_{417}]$, Dream Man $[e_{128}]$, Air Bud $[e_{348}]$, Phat Beach $[e_{439}]$. The answer is:	Candyman Correct

Table 13: An example input prompt for the Movielens dataset used by UCGRec and LLM-based model. Each $[e_n]$ indicates the embedding of item n. Gray embeddings represent simple collaborative information. Blue embeddings represent user-centric graph knowledge that integrates both temporal preferences and collaborative preferences through structural relationships.

LLM-based Prompt w/o embedding (TALLRec)	**Generated Answer**
The visit history of this user is: The Graduate, Traveller, Contact, Glory, G.I. Jane, Raiders of the Lost Ark, Sabrina, The Fifth Element, The Edge, The Devil's Own. Please predict the next movie this user will watch. Choose the one answer from the following movie titles: Shall We Dance?, Withnail and I, Striptease, Blood Beach, Young Guns, Lady of Burlesque, Grosse Pointe Blank, Pushing Hands, Clean Slate (Coup de Torchon), Dream With the Fishes, Winnie the Pooh and the Blustery Day, Murder in the First, When Harry Met Sally..., Money Train, Circle of Friends, Shadow Conspiracy, Escape to Witch Mountain, Die Hard: With a Vengeance, Waiting to Exhale, The Beverly Hillbillies. The answer is:	Dream With the Fishes Incorrect
LLM-based Prompt (LLaRA)	**Generated Answer**
The visit history of this user is: The Graduate $[e_{21}]$, Traveller $[e_{22}]$, Contact $[e_{23}]$, Glory $[e_{24}]$, G.I. Jane $[e_{25}]$, Raiders of the Lost Ark $[e_{26}]$, Sabrina $[e_{27}]$, The Fifth Element $[e_{28}]$, The Edge $[e_{29}]$, The Devil's Own $[e_{30}]$. Please predict the next movie this user will watch. Choose the one answer from the following movie titles: Shall We Dance? $[e_{58}]$, Withnail and I $[e_{121}]$, Striptease $[e_{243}]$, Blood Beach $[e_{199}]$, Young Guns $[e_{138}]$, Lady of Burlesque $[e_{305}]$, Grosse Pointe Blank $[e_{78}]$, Pushing Hands $[e_{29}]$, Clean Slate (Coup de Torchon) $[e_{268}]$, Dream With the Fishes $[e_{160}]$, Winnie the Pooh and the Blustery Day $[e_{217}]$, Murder in the First $[e_{281}]$, When Harry Met Sally... $[e_{103}]$, Money Train $[e_{41}]$, Circle of Friends $[e_{294}]$, Shadow Conspiracy $[e_{182}]$, Escape to Witch Mountain $[e_{67}]$, Die Hard: With a Vengeance $[e_{221}]$, Waiting to Exhale $[e_{307}]$, The Beverly Hillbillies $[e_{499}]$. The answer is:	Shadow Conspiracy Incorrect
Our Prompt (UCGRec)	**Generated Answer**
The user's long-term interests are Traveller $[e_{22}]$, Contact $[e_{23}]$, G.I. Jane $[e_{25}]$, The Fifth Element $[e_{28}]$, The Edge $[e_{29}]$, and short-term interests are The Fifth Element $[e_{28}]$, The Edge $[e_{29}]$, The Devil's Own $[e_{30}]$, and visit history of this user is The Graduate $[e_{21}]$, Traveller $[e_{22}]$, Contact $[e_{23}]$, Glory $[e_{24}]$, G.I. Jane $[e_{25}]$, Raiders of the Lost Ark $[e_{26}]$, Sabrina $[e_{27}]$, The Fifth Element $[e_{28}]$, The Edge $[e_{29}]$, The Devil's Own $[e_{30}]$. Please predict the next movie this user will watch. Note that both of user long-term interests and short-term interests should be holistically considered for a more comprehensive understanding of user behavior. Choose the one answer from the following movie titles: Shall We Dance? $[e_{58}]$, Withnail and I $[e_{121}]$, Striptease $[e_{243}]$, Blood Beach $[e_{199}]$, Young Guns $[e_{138}]$, Lady of Burlesque $[e_{305}]$, Grosse Pointe Blank $[e_{78}]$, Pushing Hands $[e_{29}]$, Clean Slate (Coup de Torchon) $[e_{268}]$, Dream With the Fishes $[e_{160}]$, Winnie the Pooh and the Blustery Day $[e_{217}]$, Murder in the First $[e_{281}]$, When Harry Met Sally... $[e_{103}]$, Money Train $[e_{41}]$, Circle of Friends $[e_{294}]$, Shadow Conspiracy $[e_{182}]$, Escape to Witch Mountain $[e_{67}]$, Die Hard: With a Vengeance $[e_{221}]$, Waiting to Exhale $[e_{307}]$, The Beverly Hillbillies $[e_{499}]$. The answer is:	Grosse Pointe Blank Correct