USER-LLM: User History Encoding and Compression for Multimodal Recommendation with Large Language Models

Anonymous ACL submission

Abstract

While large language models (LLMs) have proven effective in leveraging textual data for recommendations, their application to multimodal tasks involving visual content remains underexplored. Although LLM can comprehend multimodal content through a projection function that maps visual features into the semantic space of LLM, recommendation tasks often require representing users' history interactions through lengthy prompts combining text and visual elements, which not only hampers training and inference efficiency but also makes it difficult for the model to accurately capture user preferences from complex and extended prompts, leading to reduced recommendation performance. To address this challenge, we introduce USER-LLM, an innovative multimodal recommendation framework that integrates textual and visual features through a User History Encoding Module (UHEM), compressing multimodal user history interactions into a single token representation, effectively facilitating LLMs in processing user preferences. Extensive experiments demonstrate the effectiveness and efficiency of our proposed mechanism. [1]

1 Introduction

Nowadays, recommendation models have seen remarkable improvements, particularly with the rise of LLMs, which offer powerful capabilities for generalization and reasoning. LLMs have played a significant role in enhancing the performance of recommendation systems, driving a shift in the paradigm of modern recommendation approaches (Lin et al., 2023; Wu et al., 2024b).

Previous studies (Bao et al., 2023; Zhang et al., 2023) have employed LLMs in recommendation systems by presenting textual content from users' history interactions and the candidate item as prompts, allowing the LLMs to infer whether the user would prefer the given candidate item. These approaches leverage LLMs' advanced text comprehension capabilities to effectively capture user preferences and improve recommendation performance. However, for multimodal recommendation tasks, incorporating non-textual modalities, such as images and videos, into the modeling with LLMs remains relatively unexplored. A prevalent strategy of letting LLMs comprehend multimodal content involves mapping visual features into LLMs' semantic space through a projection function (Liu et al., 2024a; Li et al., 2023b). Therefore, an intuitive method for multimodal recommendation is to combine both text embedding and projected visual features in prompts, enabling LLMs to discern users' multimodal preferences and facilitate multimodal recommendations as shown in Figure 1.

In practical applications, users with long history interactions are often encountered, and prompts

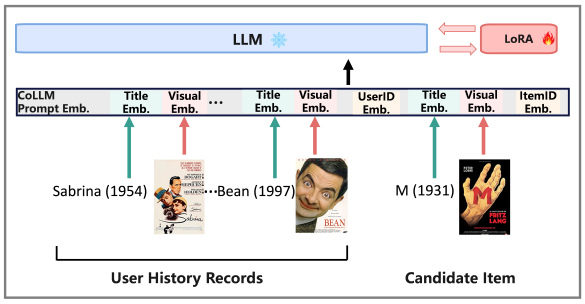

Figure 1: Incorporating Visual Features into CoLLM Embeddings. The original prompt of CoLLM and the prompt with visual features can be found in Appendix.

Dataset		movie	netflix
Scenario	**Method**	**AUC**	**AUC**
Short	CoLLM-VA	**0.8070**	**0.6725**
Long	CoLLM-VA	0.8067	0.6668

Table 1: Performance Evaluation of CoLLM-VA in Short and Long interaction Scenarios. CoLLM-VA refers to the integration of Visual Alignment into CoLLM.

[1] Once accepted, we will release our code on GitHub.

that mix long multimodal historical data present two main challenges. First, longer prompts slow down the speed of model training and inference. Second, the mixture of multimodal information in lengthy prompts increases the difficulty for the model to understand user preferences. Our preliminary experiments compared the recommendation performance for users with long and short history interactions as shown in Table 1. On average, the model performs better for users with shorter history interactions than those with longer ones, showing that LLMs lack the ability to effectively process long history interactions, which ultimately degrades recommendation performance.

To address the aforementioned issues, we propose a User History Encoding Module (UHEM) that compresses multimodal user history interactions into a single token representation in the semantic space of the LLM. This compressed user history representation is then injected into the LLM, enabling it to understand user preferences better. This approach offers two key benefits. First, it significantly reduces the length of prompts that the LLM needs to process, thereby improving training and inference efficiency. Second, it addresses the challenge faced by LLMs when dealing with extensive history interactions and long prompts. Given that UHEM can encode user history interactions of arbitrary length, this method aids the model in better understanding user preferences, especially with long history interactions, and therefore enhances recommendation performance.

Based on the ability to encode history interactions of arbitrary length, we further propose knowledge augmentation for item content to obtain richer semantic descriptions of items. This enhances the model's understanding of both items and user preferences, leading to further enhancing its recommendation capabilities. Our main contributions are summarized as follows:

- **Multimodal Recommendation**: We introduce a LLM-based multimodal recommendation framework, which integrates both text and visual modalities in a prompt design, obtaining improved recommendation performances.
- **Multimodal Encoding and Compression**: We propose UHEM to encode and compress long sequences of history interactions with both text and visual features, improving the efficiency of capturing user preferences and enhancing the model's recommendation capabilities.
- **Knowledge-Enhanced Text Representation**: To further utilize UHEM, we propose knowledge augmentation for item content to obtain richer semantic descriptions for enhancing recommendation performances.
- **Improved Recommendation Performance**: Through extensive experiments on real-world datasets, we demonstrate that our proposed method significantly outperforms existing baseline models in key performance metrics.

2 Related Work

In this section, we discuss some related work on multimodal recommendation, multimodal large language models (Multimodal LLMs), and LLM-based recommendation (LLMRec).

2.1 Multimodal Recommendation

Recent studies have explored multimodal feature integration through various approaches. Graph-based methods leverage user-item interactions, with DualGNN (Wang et al., 2021) and MMGCL (Yi et al., 2022) utilizing graph convolutions to model unimodal preferences. Item-item graphs have proven effective for representation enhancement (Mu et al., 2022; Ma et al., 2022), and MICRO (Zhang et al., 2022) combines metric learning with contrastive learning for improved multimodal fusion.

Attention mechanisms facilitate flexible multimodal integration at both coarse (Liu et al., 2021a, 2022) and fine-grained (Chen et al., 2019; Kim et al., 2022) levels. Recent works like MML (Pan et al., 2022) and MM-Rec (Wu et al., 2021) apply attention to sequence modeling and feature alignment, while VLSNR (Han et al., 2022) and NOVA (Liu et al., 2021a) employ combined attention structures for enhanced multimodal fusion.

2.2 Multimodal LLM

With the continuous development of large language models in natural language processing, more researchers are focusing on multimodal large models. In pre-training, some work aims to design improved encoders and decoders to enhance fine-grained visual perception and reasoning tasks (Wu et al., 2024a; Hao et al., 2024). The OMG-LLaVA (Zhang et al., 2024b), employs a visual encoder and integrates image information into the visual tokens

of a large language model. This enables end-to-end training of a unified encoder, decoder, and LLM, facilitating reasoning at the image, object, and pixel levels.

In fine-tuning, CityLLaVA (Duan et al., 2024) is a framework for urban scenes, combining visual cue techniques like bounding box guidance, view selection, and global-local joint views. ViP-LLaVA (Cai et al., 2024) uses eight visual cues to overlay markers on RGB images, eliminating complex region encoding and achieving state-of-the-art performance on region understanding tasks. ImageBrush (Yang et al., 2024) enables image manipulation via visual cues, reducing cross-modal differences and introducing new forms of interaction.

In instruction tuning, instruction following and structured output have been shown to enhance the capabilities of LLMs and MLLMs (Ouy; Liu et al., 2024b). AnyRef (He et al., 2024) generates pixel-level object perception by processing multimodal inputs through specialized tags and prompts, enabling consistent cross-modal reference handling.

2.3 LLMRec

The rapid development of large language models has introduced a new paradigm in recommendation algorithms. The related work can be broadly categorized into non-generative recommendation and generative recommendation based on LLMs. Non-generative recommendation aligns pre-trained large language models with recommendation tasks, receiving a list of candidate items and assigning a score or ranking to each item. For instance, LLMRec (Liu et al., 2023) benchmarks LLMs on recommendation tasks, showing better performance on interpretability than accuracy. NoteLLM (Zhang et al., 2024a) uses LLMs to generate text embeddings for item-to-item (I2I) recall, improving recommendation performance. Llama4Rec (Luo et al., 2024) combines traditional and LLM-based methods to enhance recommendation performance. RecRanker (Luo et al., 2023) fine-tunes LLMs for top-k ranking, integrating auxiliary information into prompts. ONCE (Liu et al., 2024c) uses LoRA to combine open and closed-source LLMs for content-based recommendations. TALLRec (Bao et al., 2023) integrates supplementary information while freezing original parameters. CoLLM (Zhang et al., 2023) uses LoRA and Collaborative Information Embedding Tuning (CIE) to map collaborative information into LLM inputs.

Generative recommendation creates personalized item lists based on user history. GPT4Rec (Li et al., 2023a) combines generative models and search engines, generating queries from item titles in user history to retrieve recommendations. RecGPT (Zhang et al., 2024c) uses the ChatGPT paradigm for sequential recommendation, fine-tuning an auto-regressive model with user IDs to generate personalized prompts. GenRec (Ji et al., 2024) reformats item titles based on user interactions and fine-tunes an LLM to predict the next items.

3 Method

In this section, we introduce the problem definition and the detailed architecture of our model, followed by an explanation of the fine-tuning method.

3.1 Problem Definition

Let U represent a user and I represent a candidate item. The recommendation task can be represented as (U, I, y), where $y \in \{0, 1\}$ indicates whether the user liked the candidate item. Specifically, the item I is defined as $I = (i, T_i, P_i)$, where i is the item ID, T_i represents the title of the item, and P_i denotes the item's image. Similarly, the user U is defined as $U = (u, I_u)$, where u is the user ID and $I_u = \{I_t\}_{t=1,2,\ldots,n}$ denotes the set of user's history interactions, where n being the total number of history interactions.

3.2 Model Architecture

Figure 2 illustrates the architecture of USER-LLM. Our framework is composed of four key modules: **Knowledge Enhancement**, **Visual Modality Alignment**, **User History Encoding Module** and **Collaborative Information Alignment**. The prompt, as depicted in Figure 2, is designed to effectively integrate the outputs from all these modules. Specifically, the prompt contains five placeholders:

- `<ItemDescription>` refers to the knowledge-enhanced description of the candidate item, generated by the Knowledge Enhancement Module.
- `<Image>` is the placeholder for the projected visual embedding provided by the Visual Modality Alignment Module.
- `<HistoryInteractions>` holds the embedding produced by the User History Encoding

Modul, which condenses the user's history interactions, including both textual and visual information.

- <UserID> and <ItemID> serve as placeholders for the collaborative embeddings produced by the Collaborative Information Alignment Module.

The following sections provide a detailed introduction to the model architecture.

3.2.1 Knowledge Enhancement

In our work, we choose advanced LLM to achieve knowledge enhancement. We employ FLAN-T5-XXL as an example to generate knowledge-enhanced descriptions with the original item titles, as these titles are often brief and contain limited information.

$$D_k = \text{FLAN-T5}(\text{prompt}(T_k)) \tag{1}$$

where T_k represents the original title and D_k is the knowledge-enhanced description generated by FLAN-T5-XXL. The prompt we use can be found in the Appendix. This enhancement enriches the input with more meaningful and relevant information for the recommendation task.

3.2.2 Visual Modality Alignment

This module consists of two parts: the Visual Embedding and the Mapping Module.

Visual Embedding. In our study, we leverage a pre-trained Vision Transformer model to extract image features. We choose a pre-trained dino_vits16 as an example.

$$p_k = f_\phi(P_k) \tag{2}$$

where P_k represents the image, $f_\phi(P_k)$ denotes the process of obtaining the visual representation through a pre-trained Vision Transformer model, and $p_k \in R^{1\times d_1}$ represents the visual representation with dimension d_1.

Mapping Module. For visual embeddings p_k, we apply a mapping module to project the visual feature into the LLM's semantic space:

$$\mathbf{e_{p_k}} = M_\varphi(p_k) \tag{3}$$

where $\mathbf{e_{p_k}} \in R^{1\times d_3}$ represents the projected visual embedding in the LLM's semantic space, and M_φ is the mapping module parameterized by φ.

3.2.3 User History Encoding Module

We construct item-level embeddings by concatenating the embeddings of textual descriptions with the visual projection of a given item. For a user's history interactions, these item representations are sequentially concatenated. To manage the potentially lengthy representations, we compress them into a single token embedding, which serves as a compact representation to replace the <HistoryInteractions> placeholder.

For a single item, the process can be formalized as follows:

$$s_k = \text{Tokenizer}(D_k) \quad \mathbf{e_{d_k}} = \text{Encoder}(s_k) \tag{4}$$

where D_k represents the knowledge-enhanced description of the k-th item in the user's interactions, which is processed by the built-in tokenizer and encoder of the LLM to obtain embeddings. s_k is the tokenization output, and $\mathbf{e_{d_k}}$ denotes the k-th item's description embeddings.

$$\mathbf{e_k} = \text{Concatenate}(\mathbf{e_{d_k}}, \mathbf{e_{p_k}}) \tag{5}$$

where $\mathbf{e_{p_k}}$ represents the projected visual embeddings, and $\mathbf{e_k}$ is the representation of the k-th item.

For the entire sequence of history interactions, we concatenate the representations of all items as follows:

$$\mathbf{e_{his}} = \text{Concatenate}(\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_n}) \tag{6}$$

where $\mathbf{e_{his}}$ represents the concatenated embeddings of the n items in the history interactions.

To handle the concatenated history interaction embeddings $\mathbf{e_{his}}$, we utilize a GRU (Gated Recurrent Unit) network to compress the embeddings. The GRU network captures the temporal dependencies across the items in the sequence, and the final token embedding from the GRU's output is used as the final representation of the history interactions. The process can be formalized as follows:

$$\mathbf{h_t} = G_\beta(\mathbf{e_{his}}) \tag{7}$$

where the concatenated history interaction embeddings $\mathbf{e_{his}}$ are passed through a GRU network with the parameter β. The last token embedding $\mathbf{h_t} \in R^{1\times d_3}$ is taken as the final embedding representation of the history interactions.

3.2.4 Collaborative Information Alignment

In our work, we follow the CoLLM approach (Zhang et al., 2023), which enhances the recommendation performance by incorporating collaborative filtering information.

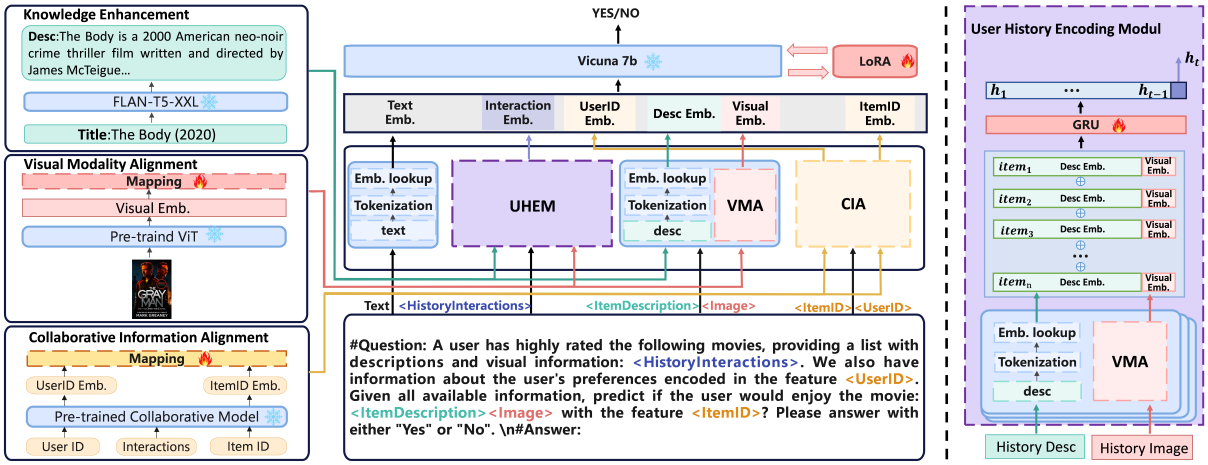

Figure 2: Model architecture overview of USER-LLM. The left part is the knowledge enhancement module, visual modality alignment module and collaborative information module. The central part is the process of LLM-based prediction. The right part is the specific details of the user history encoding module.

Collaborative Embedding. We use a pre-trained collaborative model to get the userID embedding and the itemID embedding.

$$\mathbf{u} = f_\psi(U, (U, I, y)) \quad \mathbf{i} = f_\psi(I, (U, I, y)) \quad (8)$$

where $\mathbf{u}, \mathbf{i} \in R^{1\times d_2}$ denote the user and item embeddings with dimension d_2, and $f_\psi(\cdot)$ denotes the process of obtaining representations through a pre-trained collaborative model, such as Matrix Factorization (MF).

Mapping Module. Similarly for collaborative embeddings $\mathbf{u}, \mathbf{i}$, the mapping module projects these embeddings into the LLM's semantic space:

$$\mathbf{e_u} = M_\omega(\mathbf{u}) \quad \mathbf{e_i} = M_\omega(\mathbf{i}) \quad (9)$$

where $\mathbf{e_u}, \mathbf{e_i} \in R^{1\times d_3}$ are the projected collaborative embeddings in the LLM's semantic space, and M_ω is the mapping module parameterized by ω.

3.2.5 LLM Prediction

After replacing the placeholders with embeddings, the final representation E' is fed into the LLM for inference. LoRA fine-tuning is applied to adapt the model's parameters. During LoRA fine-tuning, the original model parameters θ_{orig} are updated by adding low-rank matrices θ_{LoRA}, which represent the adaptation. The updated model parameters θ are the sum of the original parameters and the low-rank adaptation:

$$\theta = \theta_{\text{orig}} + \theta_{\text{LoRA}} \quad (10)$$

The final output of LLM can be expressed as follows:

$$\hat{y} = \text{LLM}_\theta(E') \quad (11)$$

where $\hat{y}$ is the model's predicted result. The training process minimizes the binary cross-entropy loss L, which is calculated between the true label y and the predicted probability $\hat{y}$:

$$L = -\left(y \cdot \log(\hat{y}) + (1-y) \cdot \log(1-\hat{y})\right) \quad (12)$$

3.3 Tuning Method

In our approach, we adopt a two-step fine-tuning method.

Step 1: LoRA Fine-Tuning. In the first step, we fine-tune the LLM with Lora. The optimization process for fine-tuning the LoRA parameters is as follows:

$$\theta^*_{\text{LoRA}} = \arg\min_{\theta_{\text{LoRA}}} L(y, \hat{y}) \quad (13)$$

where $L(y, \hat{y})$ is the cross-entropy loss between the true label y and the predicted output $\hat{y}$. The fine-tuning process here only updates the LoRA parameters θ_{LoRA}, while the original LLM parameters θ_{orig} remain frozen.

Step 2: Fine-Tuning the UHEM and the Mapping Modules. In the second step, we freeze the LoRA parameters θ_{LoRA} and fine-tune the UHEM and the mapping modules. The optimization for fine-tuning the mapping and compression modules can be written as follows:

$$\Theta = \arg\min_{\Theta} L(y, \hat{y}) \quad (14)$$

where $\Theta = (\varphi, \omega, \beta)$, with φ representing the parameters of the visual mapping module, ω denoting the parameters of the collaborative mapping module, and β referring to the parameters of the GRU.

dataset	movie	netflix
#User	605	803
#Item	2400	3219
#Positive	15107	21931
#Negative	5641	16808
#Poster	2381	3135
#Train	16598	30991
#Valid	2074	3873
#Test	2076	3875

Table 2: Statistics of the evaluation datasets.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on two real-world recommendation datasets, and the statistical information of the processed dataset is available in Table 2.

The Movies Dataset [2] is a large-scale dataset available on Kaggle, consisting of metadata about movies, ratings, URLs and user interactions. We selected 605 users with at least 5 history interactions. For each item, we crawled the corresponding poster from the URLs provided in the metadata. User ratings for the items range from 1 to 5. We classified ratings of 2.5 or higher as positive examples, and ratings lower than 2.5 as negative examples. In the following experiments, we will refer to The Movies Dataset as "movie".

Netflix Prize Data (Wei et al., 2023) provided posters for The Netflix Prize dataset [3], which is a collection of movie ratings data made available as part of the Netflix Prize competition. The dataset includes user-item ratings, where users rate movies on a scale from 1 to 5. We selected 803 users who had at least 10 history interactions. Ratings of 4 or higher were classified as positive examples, while ratings below 4 were considered negative examples. In the following, we will refer to the Netflix Prize Data as "netflix".

Evaluation Metrics. In our work, we primarily use two evaluation metrics: AUC and UAUC (Liu et al., 2021b). A higher AUC indicates a better-performing model in terms of its ability to distinguish between positive and negative instances. UAUC essentially evaluates how well the model can recommend items for each individual user, rather than across the entire dataset.

Compared Methods. The compared methods include both traditional recommendation models and LLM-based recommendation algorithms.

- **MF** (Koren et al., 2009): MF is a classical collaborative filtering technique widely used for recommendation tasks.
- **SASRec** (Kang and McAuley, 2018): SASRec is a sequential recommendation model that leverages the power of self-attention mechanisms to capture user-item interactions over time.
- **LightGCN** (He et al., 2020): LightGCN is a graph-based recommendation model that simplifies traditional graph convolutional networks by removing unnecessary components.
- **TALLRec** (Bao et al., 2023): TALLRec can learn not only from user-item interactions but also from the rich textual information embedded in item titles by fine-tuning the LLM.
- **CoLLM** (Zhang et al., 2023): CoLLM combines the traditional collaborative filtering methods like Matrix Factorization with the power of LLMs.

In our experiments, both TALLRec and CoLLM are fine-tuned using Vicuna 7B with LoRA. CoLLM utilizes a pre-trained MF model for collaborative filtering.

Implementation Details. Our results are based on the average of five experimental runs. To adapt the model for recommendation purposes, we fine-tune Vicuna 7B with LoRA. For knowledge enhancement, we require the descriptions generated by FLAN-T5-XXL. For collaborative embedding outputted by MF, we set the embedding dimension d_2 to 256. Meanwhile, for visual embeddings, the output dimension d_1 of dino_vits16 is 384. For the LLM-based methods, we use the AdamW optimizer with a weight decay of 1e-3. For the LoRA module, we follow the same configuration as in the TALLRec paper, setting the rank (r) to 8, the scaling factor (alpha) to 16, the dropout rate to 0.05, and the target modules to "[q_proj, v_proj]". We employ Binary Cross-Entropy (BCE) as the optimization loss for all methods. For the movie and netflix datasets, we set the number of history interactions to 5 and 10, respectively. All experiments are performed using a single NVIDIA A100 device with 80GB of memory.

[2] https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset

[3] https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data/data

Dataset	movie				netflix			
Methods	**AUC**	**AUC Imp.**	**UAUC**	**UAUC Imp.**	**AUC**	**AUC Imp.**	**UAUC**	**UAUC Imp.**
MF	0.6781	19.2%	0.6055	13.2%	0.5730	17.9%	0.5711	16.1%
SASRec	0.7464	8.8%	0.6329	8.3%	0.6499	4.0%	0.6493	2.1%
LightGCN	0.7673	5.8%	0.6405	7.0%	0.6594	2.5%	0.6414	3.4%
TALLRec	0.7219	12.4%	0.5293	29.5%	0.6382	5.9%	0.6430	3.1%
CoLLM	0.8052	0.8%	0.6690	2.4%	0.6699	0.9%	0.6613	0.2%
USER-LLM	**0.8117**	-	**0.6852**	-	**0.6757**	-	**0.6629**	-

Table 3: Overall performance comparison. "Imp." represents the relative improvement of USER-LLM over the baseline models. Bold text indicates the best results.

4.2 Performance Comparison

Table 3 provides the overall results of the performance improvements observed across five baseline models evaluated on two distinct datasets. Drawing from the results, we have the following observations:

Firstly, our proposed USER-LLM model consistently outperforms all baseline methods across both datasets, achieving the highest performance with AUC scores of 0.8117 and 0.6757 on the movie and netflix datasets respectively. This demonstrates the robust generalization capability of our approach across different recommendation scenarios.

Secondly, compared to traditional recommendation methods (MF, SASRec, and LightGCN), USER-LLM shows substantial improvements. Specifically, on the movie dataset, USER-LLM achieves relative improvements of 19.2%, 8.8%, and 5.8% on AUC over MF, SASRec, and LightGCN respectively. Similar patterns are observed on the netflix dataset, with improvements of 17.9%, 4.0%, and 2.5% respectively. This indicates that our USER-LLM framework effectively captures user preferences better than conventional approaches.

Thirdly, when comparing with LLM-based recommendation methods (TALLRec and CoLLM), USER-LLM still demonstrates superior performance. On the movie dataset, USER-LLM outperforms TALLRec by 12.4% and CoLLM by 0.8% on AUC. The netflix dataset shows similar trends with improvements of 5.9% and 0.9% respectively. Notably, LLM-based methods that solely rely on textual information, such as TALLRec, fail to outperform traditional models on several metrics, which highlights the limitations of depending exclusively on textual information.

4.3 Ablation Study

To thoroughly investigate the effectiveness of different components in our USER-LLM framework, we conduct comprehensive ablation studies. The results are presented in Tables 4, leading to several important findings:

Dataset	movie		netflix	
Methods	**AUC**	**UAUC**	**AUC**	**UAUC**
USER-LLM	**0.8117**	**0.6852**	**0.6757**	**0.6629**
w/o UHEM	0.7970	0.6613	0.6678	0.6580
w/o KE	0.8087	0.6707	0.6715	0.6607

Table 4: Results of the ablation studies over USER-LLM. KE denotes Knowledge Enhancement and UHEM stands for User History Encoding Module.

Dataset		movie		netflix	
Scenario	**Methods**	**AUC**	**Imp.**	**AUC**	**Imp.**
Short	CoLLM-VA	0.8070	0.58%	0.6725	0.48%
	USER-LLM	**0.8117**	-	**0.6757**	-
Long	CoLLM-VA	0.8067	0.63%	0.6668	1.24%
	USER-LLM	**0.8118**	-	**0.6751**	-

Table 5: Performance comparison on users with short and long history interactions. "Imp." indicates the relative performance improvement of USER-LLM compared to CoLLM-VA, which refers to incorporating Visual Alignment into CoLLM.

Comparing the full USER-LLM model with its variants, we observe that our complete framework achieves the best performance across both datasets. When removing User History Embedding Module (UHEM), we notice a performance degradation of approximately 1.8% in AUC and 3.4% in UAUC for the movie dataset, while netflix experiences similar declines of 1.2% and 0.7% respectively. This demonstrates the importance of UHEM in our framework.

When the knowledge enhancement is removed (without KE), performance decreases for both the movie dataset and the Netflix dataset, indicating that knowledge enhancement is beneficial for model performance. However, in comparison to the ablation without UHEM, the reduction in performance is less severe, suggesting that the UHEM module is more crucial.

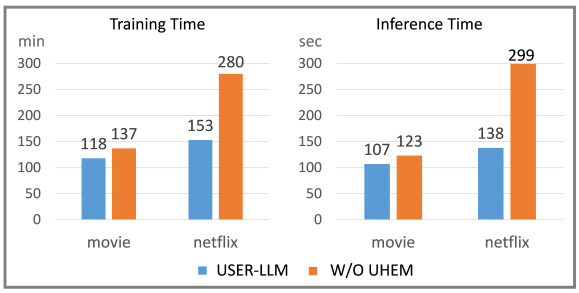

Figure 3: Comparison of computational efficiency between USER-LLM and without UHEM. The left part represents training time in Step 2, while the right part shows inference time.

4.4 Performance on Users with Short and Long History Interactions

To comprehensively evaluate our model's capability in handling users with short and long history interactions, We constructed specialized test sets to assess performance under different history interaction lengths.

For the short-history scenario, we set the number of history interactions to 5 and 10 for the movie and Netflix datasets, respectively. For the long-history scenario, we curated test sets comprising users with more extensive history interactions, specifically, users with at least 10 history interactions for the movie dataset and at least 20 for the Netflix dataset.

Table 5 presents the comparative results across these scenarios. The performance of CoLLM-VA, which incorporated visual features into CoLLM, implies that overly extended history interactions could potentially degrade recommendation effectiveness. Notably, in the long-history scenario, our model maintains robust performance with an AUC of 0.8118 on the movie dataset and 0.6751 on the netflix dataset. These results empirically validate the effectiveness of our framework in adapting to varying interaction sequence lengths.

4.5 Comparison of Computational Efficiency

To evaluate computational efficiency, we analyze both training and inference time costs. Figure 3 presents the performance comparison between USER-LLM and its variant without UHEM, which includes knowledge enhancement, visual features, and collaborative information but lacks encoded and compressed history interactions. During training (left part), USER-LLM demonstrates superior efficiency on both datasets. For the movie dataset, USER-LLM requires 118 minutes compared to 137 minutes without UHEM. This efficiency advantage is more pronounced on the Netflix dataset, where USER-LLM completes training in 153 minutes versus 280 minutes without UHEM.

The inference time comparison (right part) shows similar trends. USER-LLM achieves inference times of 107 seconds and 138 seconds on movie and Netflix datasets respectively, while the variant without UHEM requires 123 seconds and 299 seconds.

The difference in training and inference time between the two methods across different datasets indicates that when history interactions are longer (10 interactions for netflix versus 5 interactions for movie), both training and inference time increase. However, UHEM demonstrates greater improvements in computational efficiency when handling longer history interactions, as observed in the netflix dataset.

5 Conclusion

In this paper, we introduce USER-LLM, a novel multimodal recommendation framework that leverages the capabilities of LLMs to integrate multimodal data into the recommendation process. We propose UHEM, a module for encoding and compressing long sequences of history interactions with both textual and visual features into a single token representation in the semantic space of the LLM, effectively facilitating LLMs in processing user preferences. Our extensive experiments on two real-world datasets demonstrate the effectiveness of USER-LLM, achieving significant improvements in key metrics compared to existing baselines.

6 Limitations

The current framework is primarily focused on textual and visual modalities. However, the absence of other multimodal information, such as audio, may restrict the model's ability to fully grasp user preferences. Our upcoming research will concentrate on integrating additional modalities to enrich recommendation performance. Additionally, the current work employs GRU as an example encoder for UHEM, whose performance compared to other models is yet to be explored. In the future, we aim to delve into alternative encoding architectures, such as transformer-based encoders, in order to select the optimal model.

References

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.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. 2024. Vip-llava: Making large multimodal models understand arbitrary visual prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12914–12923.

Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 765–774.

Zhizhao Duan, Hao Cheng, Duo Xu, Xi Wu, Xiangxie Zhang, Xi Ye, and Zhen Xie. 2024. Cityllava: Efficient fine-tuning for vlms in city scenario. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7180–7189.

Songhao Han, Wei Huang, and Xiaotian Luan. 2022. Vlsnr: Vision-linguistics coordination time sequence-aware news recommendation. *arXiv preprint arXiv:2210.02946*.

Xixuan Hao, Wei Chen, Yibo Yan, Siru Zhong, Kun Wang, Qingsong Wen, and Yuxuan Liang. 2024. Urbanvlp: A multi-granularity vision-language pretrained foundation model for urban indicator prediction. *arXiv preprint arXiv:2403.16831*.

Junwen He, Yifan Wang, Lijun Wang, Huchuan Lu, Jun-Yan He, Jin-Peng Lan, Bin Luo, and Xuansong Xie. 2024. Multi-modal instruction tuned llms with fine-grained visual perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13980–13990.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2024. Genrec: Large language model for generative recommendation. In *European Conference on Information Retrieval*, pages 494–502. Springer.

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

Taeri Kim, Yeon-Chang Lee, Kijung Shin, and Sang-Wook Kim. 2022. Mario: modality-aware attention and modality-preserving decoders for multimedia recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 993–1002.

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

Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023a. Gpt4rec: A generative framework for personalized recommendation and user interests interpretation. *arXiv preprint arXiv:2304.03879*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, et al. 2023. How can recommender systems benefit from large language models: A survey. *arXiv preprint arXiv:2306.05817*.

Chang Liu, Xiaoguang Li, Guohao Cai, Zhenhua Dong, Hong Zhu, and Lifeng Shang. 2021a. Noninvasive self-attention for side information fusion in sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4249–4256.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Junling Liu, Chao Liu, Peilin Zhou, Qichen Ye, Dading Chong, Kang Zhou, Yueqi Xie, Yuwei Cao, Shoujin Wang, Chenyu You, et al. 2023. Llmrec: Benchmarking large language models on recommendation task. *arXiv preprint arXiv:2308.12241*.

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2024c. Once: Boosting content-based recommendation with both open-and closed-source large language models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 452–461.

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

Zhuang Liu, Yunpu Ma, Matthias Schubert, Yuanxin Ouyang, and Zhang Xiong. 2022. Multi-modal contrastive pre-training for recommendation. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*, pages 99–108.

Sichun Luo, Bowei He, Haohan Zhao, Wei Shao, Yanlin Qi, Yinya Huang, Aojun Zhou, Yuxuan Yao, Zongpeng Li, Yuanzhang Xiao, et al. 2023. Recranker: Instruction tuning large language model as ranker for top-k recommendation. *ACM Transactions on Information Systems*.

Sichun Luo, Yuxuan Yao, Bowei He, Yinya Huang, Aojun Zhou, Xinyi Zhang, Yuanzhang Xiao, Mingjie Zhan, and Linqi Song. 2024. Integrating large language models into recommendation via mutual augmentation and adaptive aggregation. *arXiv preprint arXiv:2401.13870*.

Yunshan Ma, Yingzhi He, An Zhang, Xiang Wang, and Tat-Seng Chua. 2022. Crosscbr: Cross-view contrastive learning for bundle recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1233–1241.

Zongshen Mu, Yueting Zhuang, Jie Tan, Jun Xiao, and Siliang Tang. 2022. Learning hybrid behavior patterns for multimedia recommendation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 376–384.

Xingyu Pan, Yushuo Chen, Changxin Tian, Zihan Lin, Jinpeng Wang, He Hu, and Wayne Xin Zhao. 2022. Multimodal meta-learning for cold-start sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 3421–3430.

Qifan Wang, Yinwei Wei, Jianhua Yin, Jianlong Wu, Xuemeng Song, and Liqiang Nie. 2021. Dualgnn: Dual graph neural network for multimedia recommendation. *IEEE Transactions on Multimedia*, 25:1074–1084.

Wei Wei, Chao Huang, Lianghao Xia, and Chuxu Zhang. 2023. Multi-modal self-supervised learning for recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 790–800.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Mm-rec: multimodal news recommendation. *arXiv preprint arXiv:2104.07407*.

Jiannan Wu, Muyan Zhong, Sen Xing, Zeqiang Lai, Zhaoyang Liu, Wenhai Wang, Zhe Chen, Xizhou Zhu, Lewei Lu, Tong Lu, et al. 2024a. Visionllm v2: An end-to-end generalist multimodal large language model for hundreds of vision-language tasks. *arXiv preprint arXiv:2406.08394*.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2024b. A survey on large language models for recommendation. *World Wide Web*, 27(5):60.

Yifan Yang, Houwen Peng, Yifei Shen, Yuqing Yang, Han Hu, Lili Qiu, Hideki Koike, et al. 2024. Imagebrush: Learning visual in-context instructions for exemplar-based image manipulation. *Advances in Neural Information Processing Systems*, 36.

Zixuan Yi, Xi Wang, Iadh Ounis, and Craig Macdonald. 2022. Multi-modal graph contrastive learning for micro-video recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1807–1811.

Chao Zhang, Shiwei Wu, Haoxin Zhang, Tong Xu, Yan Gao, Yao Hu, and Enhong Chen. 2024a. Notellm: A retrievable large language model for note recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 170–179.

Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Mengqi Zhang, Shu Wu, and Liang Wang. 2022. Latent structure mining with contrastive modality fusion for multimedia recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9154–9167.

Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Chen Change Loy, and Shuicheng Yan. 2024b. Omg-llava: Bridging image-level, object-level, pixel-level reasoning and understanding. *arXiv preprint arXiv:2406.19389*.

Yabin Zhang, Wenhui Yu, Erhan Zhang, Xu Chen, Lantao Hu, Peng Jiang, and Kun Gai. 2024c. Recgpt: Generative personalized prompts for sequential recommendation via chatgpt training paradigm. *arXiv preprint arXiv:2404.08675*.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. Collm: Integrating collaborative embeddings into large language models for recommendation. *arXiv preprint arXiv:2310.19488*.

A Appendix

#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, make a prediction about whether the user would enjoy the movie titled <TargetItemTitle> with the feature <TargetItemID>? Answer with "Yes" or "No". \n#Answer:

Figure 4: Prompt of enhancing the title.

#Question: A user has given high ratings to the following movies: <Item_1Title><Item_1Image>...<Item_nTitle><Item_nImage>. Additionally, we have information about the user's preferences encoded in the feature <UserID>. Using all available information, make a prediction about whether the user would enjoy the movie titled <TargetItemTitle><TargetItemImage> with the feature <TargetItemID>? Answer with "Yes" or "No". \n#Answer:

Figure 5: Prompt of enhancing the title.

#Question: Generate a concise movie description for the title <TargetItemTitle> around 20 words, highlighting the main theme and unique elements. \n#Answer:

Figure 6: Prompt of enhancing the title.

Figure 4 shows the original prompt of CoLLM. Figure 5 is the prompt that we use to incorporate visual features into CoLLM. Figure 6 shows the prompt of enhancing the title. The `<TargetItemTitle>` should be replaced by the item's title.