LLM-assisted Explicit and Implicit Multi-interest Learning Framework for Sequential Recommendation

Shutong Qiao
University of Queensland
Brisbane, Australia
shutong.qiao@uq.edu.au

Chen Gao
Beijing National Research Center for Information Science and Technology, Tsinghua University
Beijing, China
chgao96@tsinghua.edu.cn

Yong Li
Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University
Beijing, China
liyong07@tsinghua.edu.cn

Hongzhi Yin
University of Queensland
Brisbane, Australia
h.yin1@uq.edu.au

Abstract

Multi-interest modeling in current recommender systems (RS) is mainly based on user behavioral data, capturing user interest preferences from multiple dimensions. However, since behavioral data is implicit and often highly sparse, it is challenging to understand users' complex and diverse interests. Recent studies have shown that the rich semantic information in the text can effectively supplement the deficiencies of behavioral data and provide a new perspective for building more accurate user portraits. Despite this, it is still difficult for small models to directly extract semantic features associated with users' deep interests. That is, how to effectively align semantics with behavioral information to form a more comprehensive and accurate understanding of user interests has become a critical research problem.

To address this, we propose an LLM-assisted explicit and implicit multi-interest learning framework (named EIMF) to model user interests on two levels: behavior and semantics. The framework consists of two parts: Implicit Behavioral Interest Module (IBIM) and Explicit Semantic Interest Module (ESIM). The traditional multi-interest RS model in IBIM can learn users' implicit behavioral interests from interactions with items. In ESIM, we first adopt a clustering algorithm to select typical samples and design a prompting strategy on LLM to obtain explicit semantic interests. Furthermore, in the training phase, the semantic interests of typical samples can enhance the representation learning of behavioral interests based on the multi-task learning on semantic prediction and modality alignment. Therefore, in the inference stage, accurate recommendations can be achieved with only the user's behavioral data. Extensive experiments on real-world datasets demonstrate the effectiveness of the proposed EIMF framework, which effectively and efficiently combines small models with LLM to improve the accuracy of multi-interest modeling.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

Recommender System; Sequential Recommendation; Semantic Alignment; Multi-Interest Learning; Large Language Models

1 Introduction

Recommender systems (RS) [1, 3] often rely on user behavioral data to learn user preferences to provide personalized services in multiple applications of World Wide Web. Given that user interests tend to evolve over time, to more accurately capture the short-term and immediate trend of user interest changes, sequential recommendation (SR) [9, 39] has gradually become a key focus in research and applications. Different from traditional content-based [26, 28] or collaborative filtering RSs [16, 17], SR emphasizes the temporal order and contextual information of behaviors, which uses the user's past behavior sequence to model user interests and predict the items that may be of interest next. Most current SR methods model user interests as a single, comprehensive representation to capture the user's overall interests. However, users may be interested in multiple types of content over a while, so there is a clear representational bottleneck in modeling users' varying interests using a single-interest learning model.

In the MIND model [21] proposed in 2019, the concept of multi-interest learning is introduced for the first time. The model employs the dynamic routing mechanism of capsule networks to capture the diverse interests of users. Since then, the research of multi-interest learning [8, 25, 29, 41] has received increasing attention. Current multi-interest approaches mainly rely on user behavioral data to learn users' interests. However, most of this behavioral data consists of implicit feedback [37], such as clicking, browsing, and purchasing behaviors, which do not always accurately reflect the user's true preferences. It may contain noisy data [13, 30], such as records generated by mis-touching or non-true expressions of interest due to popularity bias, which may interfere with the model's understanding of the user's actual preferences. In addition, user behavioral data usually exhibits a high degree of sparsity, which further limits the model's ability to capture users' diverse and dynamically changing interests.

With the rapid development of natural language processing (NLP), the era of large language models (LLMs) [11, 27] has arrived. LLMs' outstanding language understanding, logical inference, and content generation capabilities provide new ideas and methods for solving the above challenges. That is, LLM is not only capable of extracting deep semantic information from text but also contains extensive knowledge of the outside world. With its superior contextual understanding, LLM effectively filters out irrelevant information and accurately captures user interests. Numerous studies [6, 12, 23, 42, 43] have shown that applying LLMs to RS demonstrates promising results in solving the cold-start problem and improving the interpretability of recommendation. However, since SR has high real-time requirements and the fine-tuning and inference of LLM requires a lot of time and computing resources, the fundamental problem of combining large models (LLMs) and small models (traditional algorithms) remains unsolved.

As mentioned above, multi-interest learning aims to learn the semantic-aware sub-interest, which happens to hold the same view as the LLM-enhanced RS. Therefore, combining the learned semantic information from LLMs with behavioral data is promising yet still very challenging: how to perform the alignment and how to ensure high efficiency?

To address the above challenges, we propose an LLM-assisted **E**xplicit and **I**mplicit **M**ulti-interest Learning **F**ramework (shorten as EIMF). Specifically, EIMF can be divided into two modules, the implicit behavioral interest module (IBIM) and the explicit semantic interest module (ESIM). In the training phase, IBIM follows the traditional multi-interest SR model, which can learn the implicit behavioral interests of users from their interaction data with items. ESIM, on the other hand, first analyzes user-generated text sequences using the clustering algorithm to identify representative samples. Next, these typical samples are subjected to deep inference using an LLM to reveal the users' explicit semantic interests. Finally, we match semantic and behavioral interests based on "id-cluster numbering", with joint learning on two auxiliary tasks (text classification and modal alignment), which can help fuse semantic information into behavioral interest representations. In the testing phase, accurate recommendations can be achieved based solely on user behavior data.

In summary, the contributions of this paper are as follows:

- We answer the question of how to combine large&small models in RS from the perspective of multi-interest learning and semantic alignment. We propose an LLM-assisted multi-interest learning framework, which can fully use the semantic reasoning capabilities of LLM and enrich the user's behavioral learning, thereby significantly improving the accuracy of recommendation results.
- We propose an efficient typical sample strategy to reduce the inference cost of LLM and enrich the user interest representation through multi-task learning in the training phase so that more accurate recommendations can be achieved in the testing phase by only using user behavior data.
- We conduct extensive experiments on real-world datasets, and the results show that our EIMF can significantly and stably improve the recommendation performance and exhibit excellent generalization capabilities.

2 Related Work

2.1 Sequential Recommendation

Early SR research [15, 32] focused on using the Markov Chain model to capture the transition probability between user behaviors to make recommendations. With the continuous advancement of deep learning technology, neural networks have become the mainstream method for SR. Among these methods, recurrent neural networks (RNNs) [33] are widely used due to their ability to effectively process sequence data. For example, GRU4Rec [18] uses GRU to capture changes in the user's interest in the current session. Wu et al. [38] proposed RRN, which is implemented through a long short-term memory (LSTM) autoregressive model that captures dynamics and low-rank decomposition. With the emergence of the Transformer model [36], researchers have discovered that the attention mechanism can adaptively adjust the degree of attention to each element in the input information, achieving more accurate learning of user interests. NARM [22] and SASRec [20] use the attention mechanism to model the user's sequential behavior, and capture the user's main purpose. Bert4Rec [34] performs bidirectional encoding of user sequences based on the Bert structure and combines context to predict randomly masked items.

Although these SR models perform well in performance, they usually simplify user interests into a single embedding vector, resulting in recommendation results that cannot fully reflect the user's actual multi-dimensional preferences, which in turn affects the personalization level and user experience.

2.2 Multi-interest Learning

As the research continues to deepen, researchers have gradually realized that users' interests are often diverse, and a single interest modeling method makes it difficult to capture users' complex and dynamically changing interests accurately. Alibaba's research team proposed the MIND model [21], which introduced the concept of multi-interest modeling. The model uses a dynamic routing mechanism to build a multi-interest extraction layer, which can effectively mine users' multiple interests from their historical behavior data. The ComiRec [4] model explores multi-interest learning methods using dynamic routing and self-attention mechanisms. Tan et al. proposed the SINE model [35], in which the sparse interest module can adaptively infer the sparse concept set of each user from a large concept pool, and the interest aggregation module is used to model multiple user interests. Xie et al. [40] proposed the REMI framework, which uses an interest-aware hard negative mining strategy to effectively train discriminative representations and a routing regularization method to prevent interest routing collapse. Zhu et al. [45] proposed the HPCL4SR model, in which category information was introduced into the model to construct a global graph to filter high-level preferences and used them as positive examples, and used contrastive learning to distinguish the differences between multiple interests based on user-item interaction information. PoMRec [7] first inserts specific prompts into user interactions to adapt them to the multi-interest extractor and aggregator and then utilizes the mean and variance embedding of user interactions to embed users' multiple interests.

Although the above multi-interest models can learn users' multi-dimensional interests, they only model multi-interests at the behavioral level based on the ID paradigm and do not involve the understanding and exploration of the deeper semantic meanings behind user interests.

2.3 LLM-assisted Recommender System

As LLMs continue to demonstrate their superior capabilities, more and more research has begun to explore the application of LLMs in RS. LLM-assisted RS can analyze text information related to users and items, thereby understanding user interests more deeply. ChatRec [11] combines conversational AI, such as ChatGPT, with an RS that converts user profiles and historical interactions into prompts, relying only on contextual learning for effective recommendations without training. InstructRec [44] adapts to the recommendation task by way of LLM instruction tuning. By combining the strengths of traditional CTR models with pre-trained language models, the CTRL framework [24] aims to integrate and utilize information from different modalities more effectively. E4SRec solves the problem of representing ID information by injecting the ID embedding of items in the SR model into the LLM. SAID [19] uses a projector module to convert item IDs into embedding vectors, which are fed into LLM to obtain item embeddings containing fine-grained semantic information. RLMRec [31]incorporates auxiliary text signals, uses LLM for user/item analysis, and aligns the semantic space of LLM with collaborative relationship signals through cross-view alignment.

Although LLM can significantly improve the performance of RS, LLM-assisted RS often faces challenges such as real-time requirements and large computational resources during training and deployment, which limits its feasibility for direct application in industrial environments.

3 Methodology

In Figure 1, we show the overall architecture of EIMF. In this section, we introduce the problem definition of SR in Section 3.1, then introduce EISM in Section 3.2, IBIM in Section 3.3, and finally introduce the training and serving of the framework in Section 3.4.

3.1 Problem Definition

In this paper, we define the set of all users as $\mathcal{U}$ and the set of all items as $\mathcal{I} = \{i_1, i_2, \cdots, i_N\}$, Where N is the number of unique items in the dataset. For each historical interaction sequence s_u between a user and an item set, we represent it as $S_u = [i_1^u, i_2^u, \cdots, i_n^u]$. Here n is the maximum length of the sequence, and the sequence is arranged in order according to the time of user interaction. Therefore, the task of the multi-interest SR model is to recall a subset of items from item set $\mathcal{I}$ that the user is likely to interact with next based on the user's historical behavior records. Specifically, given a user's interaction sequence $S_{u,t} = [i_1^u, i_2^u, \cdots, i_t^u]$ at the previous t steps as input, predict the item i_{t+1}^u that the user may click at the next step.

3.2 Explicit Semantic Interest Module

3.2.1 Dividing Interest Groups. Although each user's interests are highly individualized, in practical applications, those users who exhibit similar behavioral patterns are likely to share similar interests. Second, considering the relationship between data volume and LLM inference time, to effectively reduce the time required for the LLM inference process, we start from the perspective of data optimization by selecting representative data to ensure the quality of the model inference and significantly reduce the computational resource requirements.

Based on the natural similarity of text structure, we cluster the sequence data in text form to divide users into different interest groups. We chose the Affinity Propagation (AP) algorithm [10] as our clustering tool during this process. The uniqueness of this algorithm is that it determines the relationship between data points by exchanging "responsibility" and "availability" messages between data points until a stable state is reached to form clusters. This mechanism enables Affinity Propagation to dynamically determine the optimal number of clusters without pre-setting, thereby more accurately capturing the different interest patterns of users. The specific algorithm flow is shown in Algorithm 1.

$$E_t = \mathrm{BERT}(T), \tag{1}$$

where $T = [t_1, t_2, \cdots, t_n]$ represents the text form of the user's historical click sequence, BERT is a pre-trained language model.

$$C, C_{\mathrm{c}} = \mathrm{AP}(E_t, p), \tag{2}$$

where $C = \{c_1, c_2, \cdots, c_K\}$ represents the category of clustering and K is the number of clusters, and p is the hyperparameter in the AP algorithm, which can control the scale of clustering. $C_{\mathrm{c}} = \{c_{\mathrm{c}}^1, c_{\mathrm{c}}^2, \cdots, c_{\mathrm{c}}^1, c_{\mathrm{c}}^K\}$is the central sample set for each cluster, we use these central samples as typical samples for LLM inference. Note that for clusters where the center sample cannot be found, we select the closest-spaced sample as the typical sample by calculating the distance from the center value for all samples within the cluster.

3.2.2 Constructing Typical Prompts. To more effectively utilize the powerful inference capabilities of the LLM and to make it better at understanding the task, we have crafted a specific prompt. The prompt consists of three parts: 1. **Context Introduction**, 2. **User Information**, and 3. **Task Definition**. The green part in Figure 2 shows an example of a Prompt.

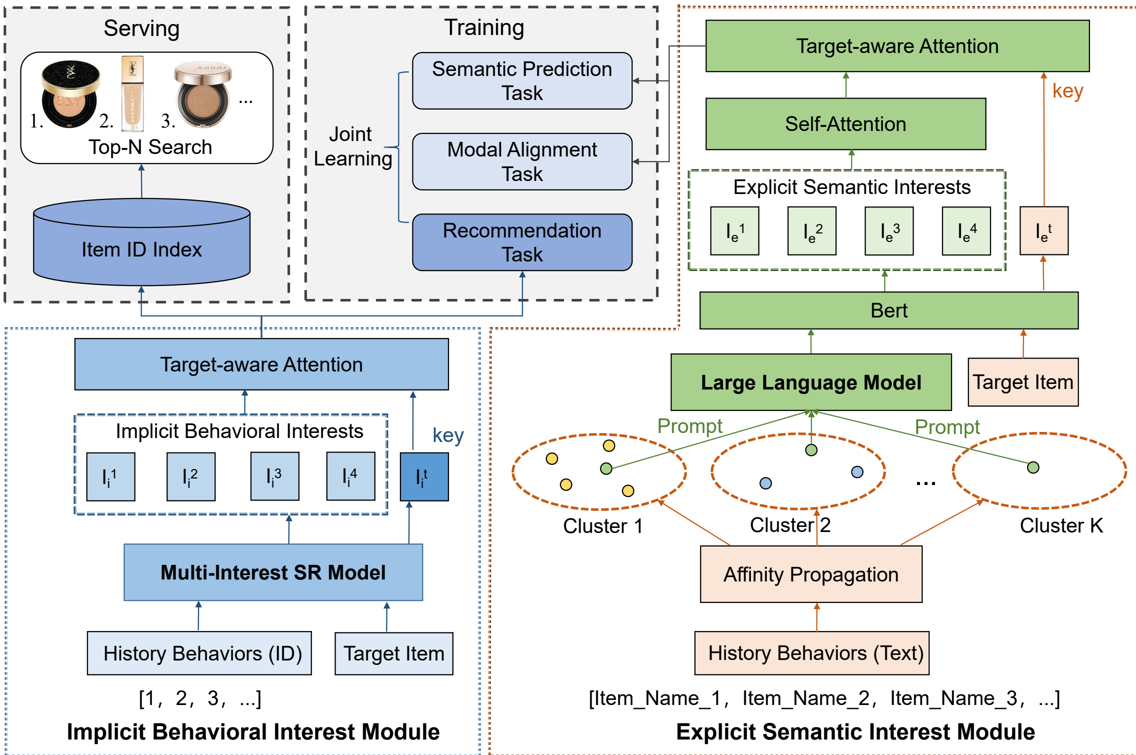


Figure 1: EIMF overall architecture diagram.

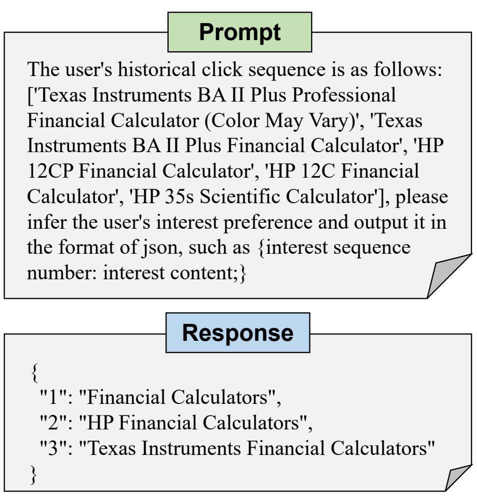


Figure 2: An example of a case that prompts design and LLM inference.

Specifically, the **Context Introduction** is a fixed template "The user's historical click sequence is as follows:", which is to help LLM understand the background meaning of the following list data; then in the **User Information** part, we use "[]" to mark the specific item list of each user, and each element in the list is designed to be in the form of an item name (ID) so that items with similar but different text names can be more distinguished in LLM; finally, the

Algorithm 1 Affinity Propagation Clustering Algorithm.

Input: $\mathbf{E_t} \in \mathbb{R}^{n \times d}$, p
Output: C, $\mathrm{C_c}$

Initialize responsibility matrix $\mathbf{R} \leftarrow \mathbf{0}_{n \times n}$
Initialize availability matrix $\mathbf{A} \leftarrow \mathbf{0}_{n \times n}$
Compute similarity matrix:
$s(i,k) = (-\|E_t[i] - E_t[k]\|^2 + p)$ for i, k in range(n)
while not converged **do**
    Update the responsibility matrix **R** using the current availability matrix **A**:
    $r(i,k) \leftarrow s(i,k) - \max_{k' \neq k}\{a(i,k') + s(i,k')\}$
    Update the availability matrix **A** using the updated responsibility matrix **R**:
    $a(k,k) \leftarrow \sum_{i' \neq k} \max\{0, r(i',k)\}$
    Determine exemplars by finding column k that maximizes $a(i,k) + r(i,k)$ for i in range(n)
end while
Assign each data point to its nearest exemplar
return C, $\mathrm{C_c}$

Task Definition part is placed at the end of the Prompt "please infer the user's interest preference and output it in the format of JSON, such as interest sequence number: interest content;", telling LLM what to do with the above information and using the JSON format to guide LLM to output the inference results.

3.2.3 Inference Semantic Interests. To capture more fine-grained semantic interests, we design an example of "serial number: interest" in prompts to guide the LLM to deep inference.

$$T_{\mathrm{r}} = \mathrm{LLM}(P_{T_c}), \tag{3}$$

where $P_{T_c} = \{p_{t_c}^1, p_{t_c}^2, \cdots, p_{t_c}^K\}$ represents the prompts corresponding to the typical samples. The user's semantic interest of each sample in T_{r} is composed of multiple sub-interests t_{r}^j.

At the same time, to distinguish each semantic interest and ensure the consistency of text representation, we use the same BERT pre-trained model as in the previous section 3.2.1 to encode each sub-interest separately, and then concatenate them to form a semantic interest representation of a typical sample.

$$h_{\mathrm{ex}} = \mathrm{Concat}(\mathrm{Bert}(t_{\mathrm{r}u}^1), \mathrm{Bert}(t_{\mathrm{r}u}^2), \cdots, \mathrm{Bert}(t_{\mathrm{r}u}^m)), \tag{4}$$

$t_{\mathrm{r}}^m \in T_{\mathrm{r}}$. $h_{\mathrm{ex}} \in \mathbb{R}^{m \times d_t}$ is the user's explicit semantic interest representation obtained by LLM inference, m represents the number of interests in LLM inference, and d_t is the text embedding dimension. For each user's semantic interest, we need to use the dictionary stored by the previous clustering algorithm in the form of "user_{id}-$(C_{\mathrm{user}_{id}}, C_c)$" to locate the corresponding typical sample semantic interest and use it as the interest representation of the user.

3.3 Implicit Behavioral Interest Module

3.3.1 Multi-interest SR Model. To more effectively mine potential interests from user behavior data represented in the form of IDs, the EIMF framework integrates the interfaces of traditional multi-interest recommendation models. We denote the set of all user ID sequences as $S = [s_1, s_2, \cdots, s_n]$,

$$H_{\mathrm{im}} = \text{Multi-Interest SR Model}(S), \tag{5}$$

where $h_{im} \in \mathbb{R}^{n_i, d}$ is the learned user implicit behavior interest, n_i represents the number of interests learned from the sequence and d is the embedding dimension.

3.3.2 Target-aware Attention Layer. The multi-interest SR model allows us to capture multiple implicit behavioral interests of a user, each of which reflects a different aspect of the user's behavioral interests. Given that the ultimate goal of the SR model is to predict the user's next click item, a target-aware attention mechanism is specifically designed. This mechanism uses target labels to screen out the most relevant potential behavioral interests during the training process. Specifically, in the target-aware attention layer, the query is the target label (e_i^T), and $k, v = h_{\mathrm{im}}$.

$$\vec{h_{\mathrm{im}}} = T_a\text{-Attention}(q, k, v), \tag{6}$$

3.4 Training & Serving

In the EIMF framework, the training phase and the serving phase are separated. During the training phase, we use a joint learning approach to leverage semantic signals inferred by the LLM to enhance the modeling of user behavior interest representations.

3.4.1 Semantic Prediction Task. Consistent with the design concept of the attention layer in the implicit behavior interest module in the previous section, we designed a two-layer attention mechanism for the explicit semantic interests derived from LLM inference. The self-attention layer aims to learn the association preferences between different semantic interests based on the semantic interests themselves; the target-aware attention layer selects the semantic interests most relevant to the semantic prediction task by injecting the text labels of the target products.

$$h_t = \text{Self-Attention}(q, k, v), \tag{7}$$

where $q, k, v = h_{\mathrm{ex}}$.

$$\vec{h_{\mathrm{ex}}} = T_a\text{-Attention}(q, k, v), \tag{8}$$

where $q = e_t^T$, $k, v = h_t$, e_t represents the text embedding of the target item name. Then, we calculate the score of user semantic interest embedding and item text embedding $\hat{y_t^k} = \mathrm{softmax}(\vec{h_{\mathrm{ex}}}^T e_t^k)$, and use cross entropy as the loss function for the semantic prediction task.

$$\mathcal{L}_{\mathrm{S}} = \sum_{k=1}^{n} y_t^k log(\hat{y_t^k}), \tag{9}$$

3.4.2 Modal Alignment Task. To ensure that the embeddings of different modalities can be in a unified embedding space, we specially designed a Modal alignment task. Specifically, the task achieves alignment between behavioral interest representation and semantic interest representation as well as alignment between item ID labels and corresponding text labels through contrastive learning and cosine similarity.

$$\mathrm{CL}(e_a, e_b) = -\frac{1}{N}\sum_{k=1}^{N} log\left(\frac{\exp(\mathrm{Sim}(e_a^k, e_b^k)/\tau)}{\sum_{j=1}^{N}\exp(\mathrm{Sim}(e_a^k, e_b^j)/\tau)}\right), \tag{10}$$

$$\mathrm{Cos}(e_a, e_b) = \frac{1}{N}\sum_{k=1}^{N}\left(1 - \mathrm{Sim}(e_a^k, e_b^k)\right), \tag{11}$$

where $\mathrm{Sim}(a, b) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}$ is the calculation method of cosine similarity, the loss function for the final modal alignment task we denote as follows,

$$\mathcal{L}_{\mathrm{A}} = \alpha(\mathrm{CL}(h_{\mathrm{ex}}, h_{\mathrm{im}}) + \mathrm{CL}(e_{\mathrm{t}}^T, e_{\mathrm{i}}^T)) + \beta(\mathrm{Cos}(h_{\mathrm{ex}}, h_{\mathrm{im}}) + \mathrm{Cos}(e_{\mathrm{t}}^T, e_{\mathrm{i}}^T)), \tag{12}$$

where α and β are hyperparameters, $\alpha + \beta = 0.5$.

3.4.3 Recommendation Task. For the main recommendation task in the training phase, we maintain the same operation as the traditional SR models. Specifically, the score of each candidate item is obtained by multiplying the user behavior interest representation and the item representation, and then softmax is applied to convert it into probability.

$$\hat{y_i^k} = \mathrm{softmax}(h_{\mathrm{ex}}^\top e_i^k), \tag{13}$$

where e_i^k is item embeddings.

$$\mathcal{L}_R = \sum_{k=1}^{n} y_i^k log(\hat{y_i^k}), \tag{14}$$

where y_i^k is ground truth. Finally, We combine the auxiliary tasks (semantic prediction, modality alignment) with the main task (recommendation) through joint learning to obtain the final loss function.

$$\mathcal{L} = \mathcal{L}_R + \gamma\left(\mathcal{L}_S + \mathcal{L}_A\right), \tag{15}$$

where γ is a hyperparameter that controls the proportion of auxiliary tasks.

Table 1: Statistics of the utilized datasets.

Datasets	Users	Items	Interactions	Avg.len.
Beauty	22,363	12,101	198,502	8.87
Grocery	14,681	8,713	151,254	10.30
Office	4,905	2,420	53,258	10.85

In the serving phase, the trained EIMF framework can be used as an extractor of user interests and only requires the user's behavior sequence without any textual input. Then, based on the multiple user interest vectors extracted by the framework, the approximate nearest neighbor approach is used to search for the top-N items with the highest similarity to these interests, to form the final set of candidate items.

4 Experiment

4.1 Experiment setup

4.1.1 Datasets and Evaluation Metrics. To evaluate the framework's performance, we selected three sub-datasets from the review dataset of the Amazon e-commerce platform[1], namely Office Products, Grocery and Gourmet Food, and All Beauty. The statistics of the three datasets after preprocessing are summarized in Table 1.

To maintain fairness, the preprocessing method of all behavioral data follows previous related studies [4, 40] and maintains a ratio of 8:1:1 for training, validation, and testing sets in our experiment. Specifically, for a single sequence, the first 80% of the item sequences are used to model user preferences, and the last 20% of the items are used as predicted labels.

In terms of metrics, we chose the Recall @K, Normalized Discounted Cumulative Gain (NDCG) @K, and Hit Ratio (HR) @K, which are commonly used in SR. K value is set to 20 and 50.

4.1.2 Implementation Details and Hyperparameter Settings. The Qwen-Turbo model [2] was chosen for the LLM in the experiments and is responsible for inference about user interests. All models were implemented in Pytorch 1.10.0 and Python 3.9 in the conda environment. We follow previous research[40], and in our experiments, the batch size = 128, the dimension = 64, and the maximum number of training iterations for all models is 1 million. The number of interests for the multi-interest model is set to 4, the number of user interests inferred by LLM to at most 20 and the optimizer is trained using Adam with the learning rate set to 0.001. The hyperparameters α and β in the Modal Representation Alignment task were set to 0.4 and 0.1, respectively, and the loss function in the auxiliary task parameter γ was set to 0.1. Please note that except for the hyperparameter experiments, we use p = -10 as the reference value setting for the AP algorithm in the rest of the experiments (clustered typical samples account for about 3% of the dataset).

4.1.3 Baselines. Regarding baseline selection, we mainly select from single-interest and multi-interest aspects, including single-interest models [5, 18, 20, 34], LLM-based RS model [14], and multi-interest models [4, 21, 40].

- Pop. An algorithm that makes recommendations based on product popularity.
- DNN [5]. The twin-tower DNN model developed by the YouTube team pools user behaviors and then uses MLP to obtain user interest representations.
- GRU4Rec [18]. A Classic SR model based on Gated Recurrent Unit (GRU).
- SASRec [20]. An SR model based on the self-attention mechanism.
- Bert4Rec [34]. An SR model based on the Bert architecture, which uses pre-trained language model technology to process user behavior sequences and captures contextual information in the sequence through a bidirectional encoder.
- LLM2Bert4Rec [14]. A framework to enhance existing SR models by leveraging semantically rich item representations provided by LLM.
- MIND [21]. The first model proposes the concept of a multi-interest model, using capsule networks to capture the user's multiple interests.
- ComiRec-SA [4]. A model that proposes diversity control based on MIND and uses the self-attention mechanism to model multi-interests.
- REMI [40]. A general multi-interest candidate matching enhancement framework including interest-aware hard negative mining strategy and routing regularization method.

4.2 Performance Study

4.2.1 Performance comparison experiment with baselines. We added the EIMF framework to REMI to conduct performance comparison experiments with other baselines. The results are shown in Table 2. Here we marked the best performance value among all models in bold and the best performance value in the baseline with an underline. Based on the results in Table 2 and Table 3, we can make the following observations:

- In the large-scale data set in Table 2, the performance of the multi-interest model is significantly better than that of the single-interest model, and as the amount of data increases, this advantage becomes more and more significant. Our analysis shows that multi-interest learning is more suitable for data-rich environments because it is able to capture more complex and diverse underlying interests among users or items, thereby more accurately understanding subtle differences between individuals.
- In the small-scale dataset shown in Table 3, the traditional single-interest SR model outperforms the multi-interest model. Our analysis shows that when the amount of data is limited, it is difficult to accurately capture the diverse interests of users by simply relying on user behavior information. Therefore, a model that focuses on a single major interest can utilize the limited information in a more concentrated manner, thus providing more accurate recommendation results.
- In the large datasets, EIMF(REMI) achieved the best performance; while in the small dataset, LLM2Bert4Rec and EIMF(REMI) performed best. First, this shows that the rich semantic knowledge in the LLM can effectively enhance the behavioral modeling ability of user representation, showing the great potential of LLM in recommendation tasks. Second, although some metrics of EIMF(REMI) are slightly lower than LLM2Bert4Rec on small datasets, its better performance on large datasets proves that

[1]https://jmcauley.ucsd.edu/data/amazon/links.html

Table 2: The result of performance comparison between baselines and EIMF on large datasets (Grocery and Beauty).

Dataset	Model	Pop	DNN	GRU4Rec	SASRec	Bert4Rec	LLM2Bert4Rec	MIND	ComiRec-SA	REMI	EIMF(REMI)	Improv.(%)
Grocery	Recall@20	0.0729	0.1252	0.1387	0.1536	0.1544	0.1259	0.1445	0.1122	<u>0.1617</u>	**0.1758**	+8.71
	Recall@50	0.1305	0.2044	0.2300	0.2508	0.2372	0.2080	0.2162	0.2076	<u>0.2574</u>	**0.2704**	+5.05
	NDCG@20	0.0448	0.0804	0.0905	0.1041	<u>**0.1074**</u>	0.0793	0.0844	0.0706	0.0953	0.1025	-4.56
	NDCG@50	0.0624	0.0969	0.1097	<u>0.1139</u>	0.1130	0.0954	0.0967	0.0918	0.1108	**0.1181**	+3.69
	HR@20	0.1252	0.2076	0.2321	<u>0.2586</u>	0.2614	0.2164	0.2328	0.1851	0.2539	**0.2750**	+6.34
	HR@50	0.2137	0.3227	0.3649	0.3750	0.3608	0.3294	0.3322	0.3220	<u>0.3832</u>	**0.3955**	+3.21
Beauty	Recall@20	0.0452	0.1613	0.1388	0.1495	0.1430	0.1383	0.1563	0.1582	<u>0.2189</u>	**0.2323**	+6.12
	Recall@50	0.0660	0.2361	0.2109	0.2220	0.2050	0.2053	0.2384	0.2594	<u>0.3420</u>	**0.3636**	+6.31
	NDCG@20	0.0213	0.0886	0.0798	0.0854	0.0846	0.0788	0.0787	0.0854	<u>0.1139</u>	**0.1204**	+5.71
	NDCG@50	0.0270	0.0932	0.0844	0.0879	0.0852	0.0823	0.0900	0.1004	<u>0.1304</u>	**0.1348**	+3.37
	HR@20	0.0675	0.2401	0.2141	0.2320	0.2221	0.2181	0.2203	0.2325	<u>0.3111</u>	**0.3268**	+5.04
	HR@50	0.0966	0.3299	0.2982	0.3129	0.2986	0.2919	0.3272	0.3545	<u>0.4532</u>	**0.4802**	+5.95

*Note that we did not use the pre-trained version of Bert4Rec here, but adopted the same training method as other models.

Table 3: The result of performance comparison between baselines and EIMF on small dataset (Office).

Model	Recall@20	NDCG@20	HR@20	Recall@50	NDCG@50	HR@50
GRU4Rec	0.0800	0.0537	0.1384	0.1770	0.0809	0.2811
SASRec	<u>0.1152</u>	0.0643	0.1792	0.1963	0.0805	0.2871
Bert4Rec	0.1037	0.0635	0.1812	0.1920	0.0834	0.3116
MIND	0.0915	0.0513	0.1466	0.1593	0.0706	0.2505
ComiRec-SA	0.0788	0.0456	0.1202	0.1589	0.0653	0.2321
REMI	0.1072	0.0637	0.1751	0.2018	0.0825	0.2973
LLM2Bert4Rec	0.1085	<u>**0.0660**</u>	<u>**0.1812**</u>	<u>0.2117</u>	<u>**0.0939**</u>	<u>0.3360</u>
EIMF(REMI)	**0.1222**	0.0649	0.1772	**0.2296**	0.0930	**0.3503**
Improv.(%)	+6.08	-1.66	-2.20	+8.45	-0.95	+4.25

Table 4: Performance experimental results of different backbones on Beauty and Office datasets.

Dataset	Beauty			Office		
Model	Recall@50	NDCG@50	HR@50	Recall@50	NDCG@50	HR@50
Bert4Rec	0.2050	0.0852	0.2986	0.1920	0.0834	0.3116
EIMF(Bert4Rec)	0.2155	0.0854	0.3071	0.2101	0.0909	0.3380
Improv.(%)	+5.12	+0.23	+2.84	+9.42	+8.99	+8.47
SASRec	0.2220	0.0879	0.3129	0.1964	0.0806	0.2872
EIMF(SASRec)	0.2499	0.0999	0.3469	0.2088	0.0878	0.3198
Improv.(%)	+12.56	+13.65	+10.86	+6.31	+8.93	+11.35
MIND	0.2384	0.0900	0.3272	0.1594	0.0707	0.2505
EIMF(MIND)	0.2518	0.0968	0.3424	0.2023	0.0793	0.2953
Improv.(%)	+5.62	+7.55	+4.64	+26.91	+12.16	+17.88

EIMF(REMI) not only has good stability but also can adapt to more diverse data environments.

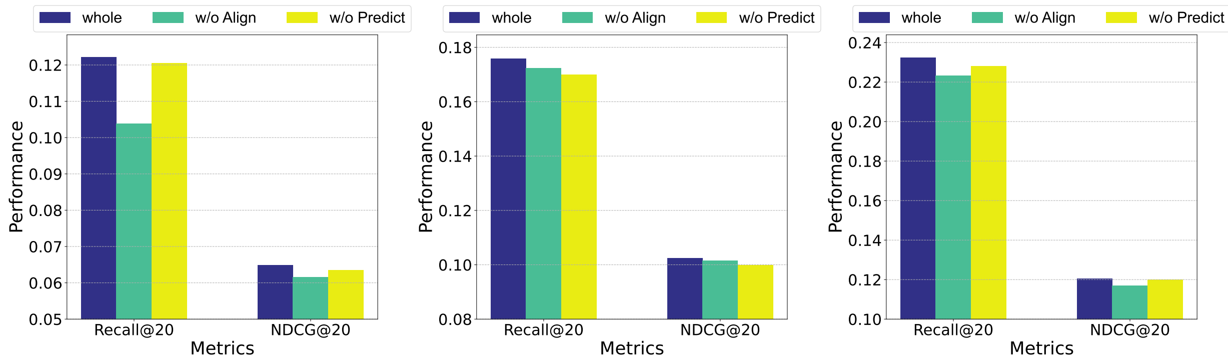
4.2.2 Performance experiment on different backbones. To explore whether the EIMF framework is compatible, we designed the following experiment. We selected three classic models as backbone, which include the traditional single-interest SR models SASRec and Bert4Rec, and the multi-interest model MIND. We compared the recommendation performance of the above models with the model combined with the EIMF framework on the Beauty and Office datasets. The experimental results are shown in the Table4.

As can be seen from Table 4, the performance of all backbone models has been improved after adding the EIMF framework, which proves the wide applicability of the EIMF framework. Our analysis suggests that this is because EIMF greatly enriches the information resources on which behavioral interest modeling relies by introducing semantic interests, effectively compensating for the limitations due to data sparsity in traditional modeling approaches. In addition, EIMF adopts an inference mechanism based on typical samples, which not only greatly reduces the amount of data for LLM reasoning tasks but also further explores and demonstrates the positive impact of group interests on individuals.

4.3 Ablation Study

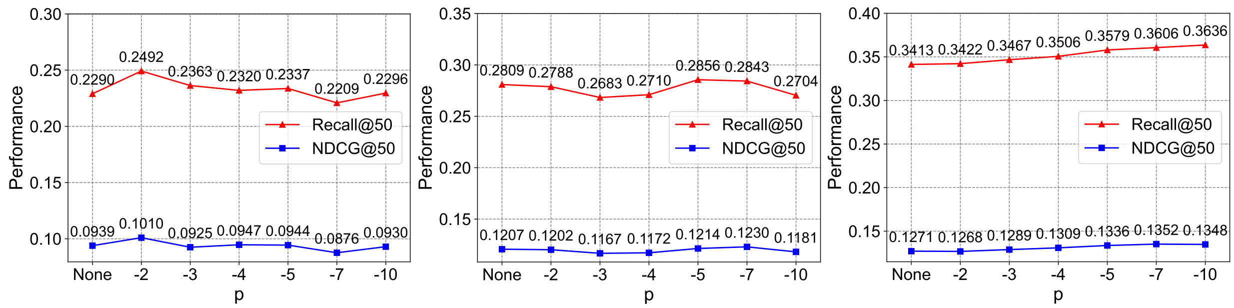
To verify the effectiveness of framework components, we designed two variants, **EIMF w/o Align** and **EIMF w/o Predict**, which represent the deleted modality alignment task and deleted semantic prediction task respectively. We experimented with the two variants and the whole framework on three datasets, and the results are shown in Figure 3.

As can be seen from Figure 3, the whole EIMF is the best-performing model on all three datasets, and the other two variants have varying degrees of performance degradation, which proves that the



(a) Office dataset (b) Grocery dataset (c) Beauty dataset

Figure 3: The results of ablation experiments on three datasets.



(a) Office dataset (b) Grocery dataset (c) Beauty dataset

Figure 4: The results of the AP clustering algorithm with different p hyperparameter settings on three datasets.

two auxiliary tasks we designed can effectively enhance the user's multi-interest representation. In addition, the performance of the other two variants drops to different degrees on different datasets. In the Office and Beauty datasets, the performance of EIMF w/o Align drops the most, while in the Grocery dataset, the performance of EIMF w/o Predict drops the most. We believe that this is because different categories of items correspond to different user behavioral patterns, e.g., when purchasing office supplies or beauty products, users' interests tend to change frequently; while when purchasing daily groceries, users' interests are more purposive. Therefore, in the Grocery dataset, the invisible interests reflected by the behavioral patterns are more homogeneous, and the semantic prediction auxiliary task dominated by explicit interests can identify more potential interests, on the contrary, in the Beauty and Office datasets, the behavioral patterns have been able to reflect the diversified interests of the users, and therefore it is more necessary to enhance the invisible user behavioral interests by using the modal alignment task. The two auxiliary tasks complement each other and enable EIMF to adapt to more scenarios.

4.4 Hyperparameter Study

We conducted hyperparameter experiments on three datasets to explore the impact of preference degree (p) on the framework recommendation performance in the AP clustering algorithm. The p can control the number of clusters in the AP algorithm. The smaller the p, the fewer clusters there are. The value of p is set to $\{None, -2, -3, -4, -5, -7, -10\}$, where $None$ means not using a clustering algorithm to infer the interests of each user individually. In addition, corresponding to $p = \{-2, -5, -7, -10\}$, the typical number of samples obtained from clustering about the proportion of the total number of data are $\{80\%, 50\%, 20\%, 5\%\}$, respectively.

It can be seen from Figure 4 that the best p-values are different on different datasets. Specifically, on the Office dataset, when $p = -2$, the Recall@50 of EIMF(REMI) is 0.2492, and the NDCG@50 is 0.1010; on the Grocery dataset, when $p = -5$, the Recall@50 is 0.2856, and the NDCG@50 is 0.1214; on the Beauty dataset, when $p = -10$, the Recall@50 is 0.3636, and the NDCG@50 is 0.1348. Through analysis and discussion, we believe that the optimal p-value selection is closely related to the amount of data and data distribution. Specifically, when the amount of data is small or the behavioral similarity between users is weak, a larger p-value helps to identify as many typical samples as possible and ensure the independence and uniqueness of user interests; when the amount of data is large or the behavioral similarity between users is high, a smaller p-value can not only effectively reduce the computational burden faced by LLM when processing large amounts of data, but also help to discover a wider range of group characteristics and enrich user interest modeling.

5 Conclusions and Future Work

In this paper, we answer the question of how to combine large&small models in RS from the perspective of multi-interest learning and semantic alignment. We propose an LLM-assisted framework with explicit and implicit multi-interest learning to capture users' interests from both behavioral and semantic levels. Specifically, the framework consists of an implicit behavioral interest module and an explicit semantic interest module. The implicit behavioral interest module adopts the traditional multi-interest RS model and learns user behavioral interest patterns by analyzing the interaction data between users and items; the explicit semantic interest module, in the training phase, first classifies users based on the AP algorithm to select typical samples of each category, and then uses LLM to infer the typical samples to obtain multiple semantic interests. These semantic interests are subsequently used to augment the representation of user behavioral interests through auxiliary tasks, including semantic prediction and modality alignment. Observed from extensive experiments on real-world datasets, our EIMF framework can effectively and efficiently combine LLM with RS models, significantly improving the recommendation performance. This two-layer interest modeling method not only considers the interest preferences directly expressed by users but also deeply explores their potential tendencies, which can provide users with more comprehensive and accurate personalized recommendations. In future research, we plan to combine multimodal and multi-behavioral data further to explore the more fine-grained user interests reflected in these data.

References

[1] Charu C Aggarwal et al. 2016. *Recommender systems*. Vol. 1. Springer.
[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).
[3] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems* 46 (2013), 109–132.
[4] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.
[5] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.
[6] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084* (2022).
[7] Xue Dong, Xuemeng Song, Tongliang Liu, and Weili Guan. 2024. Prompt-based Multi-interest Learning Method for Sequential Recommendation. *arXiv preprint arXiv:2401.04312* (2024).
[8] Yingpeng Du, Ziyan Wang, Zhu Sun, Yining Ma, Hongzhi Liu, and Jie Zhang. 2024. Disentangled Multi-interest Representation Learning for Sequential Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 677–688.
[9] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–42.
[10] Brendan J Frey and Delbert Dueck. 2005. Mixture modeling by affinity propagation. *Advances in neural information processing systems* 18 (2005).
[11] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).
[12] 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*. 299–315.
[13] Yongqiang Han, Hao Wang, Kefan Wang, Likang Wu, Zhi Li, Wei Guo, Yong Liu, Defu Lian, and Enhong Chen. 2024. Efficient Noise-Decoupling for Multi-Behavior Sequential Recommendation. In *Proceedings of the ACM on Web Conference 2024*. 3297–3306.
[14] 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*. 1096–1102.
[15] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.
[16] 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*. 639–648.
[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
[18] B Hidasi. 2015. Session-based Recommendations with Recurrent Neural Networks. *arXiv preprint arXiv:1511.06939* (2015).
[19] Jun Hu, Wenwen Xia, Xiaolu Zhang, Chilin Fu, Weichang Wu, Zhaoxin Huan, Ang Li, Zuoli Tang, and Jun Zhou. 2024. Enhancing sequential recommendation via llm-based semantic embedding learning. In *Companion Proceedings of the ACM on Web Conference 2024*. 103–111.
[20] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
[21] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2615–2623.
[22] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.
[23] Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023. GPT4Rec: A generative framework for personalized recommendation and user interests interpretation. *arXiv preprint arXiv:2304.03879* (2023).
[24] Xiangyang Li, Bo Chen, Lu Hou, and Ruiming Tang. 2023. Ctrl: Connect tabular and language model for ctr prediction. *CoRR* (2023).
[25] Yaokun Liu, Xiaowang Zhang, Minghui Zou, and Zhiyong Feng. 2024. Attribute Simulation for Item Embedding Enhancement in Multi-interest Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 482–491.
[26] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook* (2011), 73–105.
[27] OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt. (Accessed on 01/12/2023).
[28] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*. Springer, 325–341.
[29] Haolei Pei, Yuanyuan Xu, Yangping Zhu, and Yuan Nie. 2024. RimiRec: Modeling Refined Multi-interest in Hierarchical Structure for Recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*. 746–749.
[30] Shutong Qiao, Wei Zhou, Junhao Wen, Hongyu Zhang, and Min Gao. 2023. Bi-channel Multiple Sparse Graph Attention Networks for Session-based Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2075–2084.
[31] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*. 3464–3475.
[32] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.
[33] Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404 (2020), 132306.
[34] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.
[35] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*. 598–606.
[36] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[37] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*. 373–381.

[38] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 495–503.

[39] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.

[40] Yueqi Xie, Jingqi Gao, Peilin Zhou, Qichen Ye, Yining Hua, Jae Boum Kim, Fangzhao Wu, and Sunghun Kim. 2023. Rethinking multi-interest learning for candidate matching in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 283–293.

[41] Jing Yan, Liu Jiang, Jianfei Cui, Zhichen Zhao, Xingyan Bin, Feng Zhang, and Zuotao Liu. 2024. Trinity: Syncretizing Multi-/Long-Tail/Long-Term Interests All in One. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6095–6104.

[42] Mengyuan Yang, Mengying Zhu, Yan Wang, Linxun Chen, Yilei Zhao, Xiuyuan Wang, Bing Han, Xiaolin Zheng, and Jianwei Yin. 2024. Fine-Tuning Large Language Model Based Explainable Recommendation with Explainable Quality Reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9250–9259.

[43] An Zhang, Leheng Sheng, Yuxin Chen, Hao Li, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2023. On generative agents in recommendation. *arXiv preprint arXiv:2310.10108* (2023).

[44] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001* (2023).

[45] Zizhong Zhu, Shuang Li, Yaokun Liu, Xiaowang Zhang, Zhiyong Feng, and Yuexian Hou. 2024. High-level preferences as positive examples in contrastive learning for multi-interest sequential recommendation. *World Wide Web* 27, 2 (2024), 21.