LLaRA: Large Language-Recommendation Assistant

Jiayi Liao
ljy0ustc@mail.ustc.edu.cn
University of Science and Technology
of China
Hefei, China

Sihang Li
sihang0520@gmail.com
University of Science and Technology
of China
Hefei, China

Zhengyi Yang
yangzhy@mail.ustc.edu.cn
University of Science and Technology
of China
Hefei, China

Jiancan Wu
wujcan@gmail.com
University of Science and Technology
of China
Hefei, China

Yancheng Yuan
yanchengyuanmath@gmail.com
The Hong Kong Polytechnic
University
Hong Kong, China

Xiang Wang*
xiangwang1223@gmail.com
University of Science and Technology
of China
Hefei, China

Xiangnan He*
xiangnanhe@gmail.com
University of Science and Technology
of China
Hefei, China

ABSTRACT

Sequential recommendation aims to predict users' next interaction with items based on their past engagement sequence. Recently, the advent of Large Language Models (LLMs) has sparked interest in leveraging them for sequential recommendation, viewing it as language modeling. Previous studies represent items within LLMs' input prompts as either ID indices or textual metadata. However, these approaches often fail to either encapsulate comprehensive world knowledge or exhibit sufficient behavioral understanding. To combine the complementary strengths of conventional recommenders in capturing behavioral patterns of users and LLMs in encoding world knowledge about items, we introduce **L**arge **La**nguage-**R**ecommendation **A**ssistant (**LLaRA**). Specifically, it uses a novel hybrid prompting method that integrates ID-based item embeddings learned by traditional recommendation models with textual item features. Treating the "sequential behaviors of users" as a distinct modality beyond texts, we employ a projector to align the traditional recommender's ID embeddings with the LLM's input space. Moreover, rather than directly exposing the hybrid prompt to LLMs, a curriculum learning strategy is adopted to gradually ramp up training complexity. Initially, we warm up the LLM using text-only prompts, which better suit its inherent language modeling ability. Subsequently, we progressively transition to the hybrid prompts, training the model to seamlessly incorporate the behavioral knowledge from the traditional sequential recommender into the LLM. Empirical results validate the effectiveness of our proposed framework. Codes are available at https://github.com/ljy0ustc/LLaRA.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Sequential Recommendation, Large Language Models, Curriculum Learning, Hybrid Prompting

1 INTRODUCTION

Sequential recommendation [12, 48] is to predict users' next items of interest based on their historical interactions with items. Conventional sequential recommenders [16, 25, 44] typically involve two steps: (1) assigning each item with a distinct ID, which is converted into a trainable embedding; (2) learning these embeddings with the objective of next item prediction, so as to capture user preference. After training on historical interaction data, item representations can encapsulate the sequential behavioral patterns of users.

Recently, inspired by the great success of Large Language Models (LLMs) [4, 6, 45], exploring the potential of LLMs in sequential recommendation is attracting attention [2, 8, 9, 13, 20, 28, 32, 55, 58], especially driven by extensive world knowledge and innate reasoning capabilities of LLMs. At the core is to reshape sequential recommendation as the language modeling task — that is, convert the behavioral sequence into the textual input prompt, *e.g.,* "This
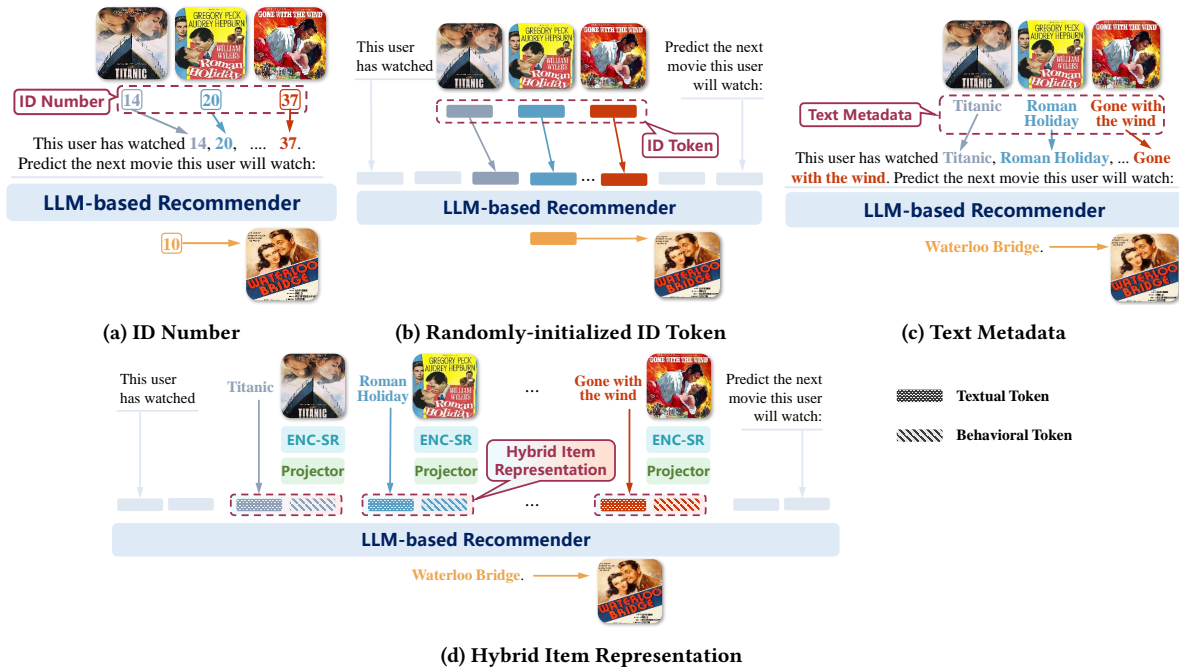
Figure 1: Comparison among three prior item representation methods and ours. (a) ID Number: represents an item with a numerical index. (b) Randomly-initialized ID Token: represents an item with an OOV-independent token. (c) Text Metadata: represents an item with its textual features, such as item title. (d) Hybrid Item Representation: integrates both textual tokens and behavioral tokens derived from the ID-based item embedding learned by traditional recommender models.

user has watched [item$_1$], [item$_2$], ..., [item$_n$]. Predict the next movie this user will watch.". When considering the way to represent the item within the prompt (*e.g.,* [item$_k$]), prior studies generally follow two approaches:

- **ID-based Representation:** Within the prompt, each item is represented as an ID number [13] (*e.g.,* "14" for the movie "Titanic") or a randomly-initialized ID token [22], as Figures 1a and 1b illustrate, respectively. Despite its simplicity, this approach leaves the textual characteristics of items (*e.g.,* titles and descriptions) untouched, consequently underutilizing the world knowledge inherent in LLMs. Moreover, the employment of ID numbers or ID tokens might pose integration challenges with LLMs, as it does not correspond well with the natural language processing capabilities of LLMs.
- **Text-based Representation:** This approach encodes each item in the prompt through its textual metadata, such as titles [2, 8] and descriptions [18, 28]. Taking Figure 1c as an example, the movie can be directly represented by its title "Titanic". While effectively harnessing LLMs' linguistic capabilities and world knowledge about items, it falls short of exhibiting the sequential behavior patterns of users. Overlooking such patterns could confine LLM in a suboptimal position when predicting the next item.

Consequently, we argue that merely prompting LLMs with either ID-based or text-based representations of item sequences fails to fully tap into LLMs' potential for sequential recommendation. Instead, the LLMs should gain a deeper understanding of the behavioral patterns inherent in the sequential interactions.

In pursuit of this goal, we explore the alignment between LLMs and the sequential recommenders, going beyond relying on mere ID-based or text-based prompting. Drawing inspiration from Multimodal Large Language Models (MLLMs) [1, 11, 23, 59] that adeptly understand and reason across diverse modalities (*e.g.,* images, audio, and 3D point clouds), we propose viewing the "sequential behaviors of users" as a new modality for LLMs in recommendation and aligning it with the language space. Such an alignment could empower LLMs to understand and internalize the behavioral patterns that recommenders have effectively identified and utilized.

To this end, we propose a novel framework as illustrated in Figure 2a, named **L**arge **La**nguage-**R**ecommendation **A**ssistant (**LLaRA**), which integrates conventional sequential recommenders into LLMs with two tailor-made enhancements:

(1) Hybrid Prompt Design: We exploit two distinct approaches, text-only and hybrid prompting, to convert an interaction sequence into an input prompt for LLMs. Specifically, the text-only method represents each item using its textual metadata, which are then transformed into textual tokens. Beyond text-only prompting, we further devise hybrid prompting, which integrates behavioral patterns sourced from recommenders. That is, for an item's ID representation from a traditional recommender (*e.g.,* SASRec [25]), we feed it into a projector (*e.g.,* a trainable MLP) to yield a behavioral token that is compatible with the LLMs' textual token space. We then combine the textual and behavioral tokens, creating a multifaceted representation of each item within the prompt. Considering the movie example in Figure 1d, [item$_1$] is depicted as the concatenation of textual token ▦ of word "Titanic" and behavioral

token ▨▨▨▨. Such an integration offers a more holistic depiction of user behaviors, surpassing prompts solely based on the ID or text.

(2) Curriculum Prompt Tuning: Building upon the dual prompting approaches, we draw inspiration from curriculum learning [3, 49] and propose a curriculum prompt tuning strategy — gradually shifting the learning focus from text-only prompting to hybrid prompting. Specifically, our strategy begins with text-only prompting, serving as an initial warm-up phase for the LLM. This phase is designed to align with the natural language modeling capabilities of the LLM, as it involves characterizing items through their textual metadata. The tuning in this phase ensures that the LLM becomes acquainted with the basic idea of the recommendation mechanism. Following this, we transition to hybrid prompting, which trains the projector to inject behavioral knowledge from recommenders into the LLM effectively.

Overall, we not only familiarize the LLM with the recommendation mechanism utilizing text-only prompts, but also internalize the behavioral knowledge encoded by recommenders with hybrid prompts. The progressive tuning strategy ensures an evolving learning experience for the LLM, enhancing its capabilities of sequential recommendation with a deeper understanding of user behavior.

We conduct experiments on three datasets, MovieLens [14], Steam [25], and LastFM [5], to compare LLaRA with various leading sequential recommender models and several LLM-based methods. The results show that LLaRA consistently outperforms these baselines in terms of the HitRatio@1 metric, demonstrating its superiority. Furthermore, we perform ablation studies to justify the importance of the two key components: hybrid prompting and curriculum prompt tuning.

In summary, our contributions can be concluded as follows: We propose a novel framework, LLaRA, to enhance LLMs with sequential recommenders. In LLaRA, we introduce a hybrid prompting method that integrates both world knowledge and behavioral patterns into item representations; and we conduct curriculum prompt tuning to achieve modality alignment. Comprehensive experimental results underscore the effectiveness of the LLaRA framework.

2 RELATED WORK

In this section, we provide a literature review pertaining to Large Language Models, Multi-modal Large Language Models, and LLMs for Sequential Recommendation. Our work draws inspiration from them for fusing LLMs and sequential recommendation systems.

2.1 Large Language Models

Language modeling has been extensively scrutinized for language understanding and generation over the past years, thereby catalyzing the recent emergence of Language Models (LMs) [4, 6, 10, 40, 45]. Pretrained LMs built on the Transformer architecture, such as BERT [10] and T5[40], have demonstrated profound versatility owing to their large-scale training corpus. More recently, researchers have delved deeper into the scaling effect by augmenting the parameter and training corpus scale to an unprecedented magnitude — encompassing billions of parameters and trillions of training tokens. These Large Language Models (LLMs), like GPT-4 [36] and Llama [45], manifest substantial performance enhancements and show emergent abilities, such as commonsense reasoning and instruction

following. Moreover, domain-specific LLMs, such as those in the domain of finance [51], medicine [41], and law [7], are constructed by integrating domain expertise with the commonsense knowledge inherent in general LLMs. These advancements inspire us to probe the potential of LLMs in the domain of recommendation.

2.2 Multi-Modal Large Language Models

Despite their versatility and promising performance, most LLMs are restricted to textual inputs. However, a vast reservoir of information and knowledge resides in other modalities, including vision, video, and audio. Consequently, researchers have proposed Multi-modal Large Language Models (MLLMs), to integrate the text with other modalities [30, 39]. Recent MLLMs suggest that visual space can be harmoniously aligned with textual space [11, 27, 47, 59], thereby empowering them to perform language generation tasks conditioned on visual inputs. Beyond vision, other modalities like video [56], audio [35], graph [34], and 3D point clouds [17, 29] are incorporated into LLMs, enabling them to digest information and knowledge of other modalities. We draw inspiration from these prior studies to devise LLaRA, which fuses multi-modal information to enhance sequential recommendation.

2.3 LLMs for Sequential Recommendation

Sequential recommendation aims to predict the next item that matches user preference, based on his/her historical interaction sequence [12, 48]. Prior studies have explored employing complex model architectures to better characterize user preference, including Recurrent Neural Networks (RNNs) [16, 38, 43], Convolutional Neural Networks (CNNs) [44, 53], and Attention mechanisms [25, 42]. With the advent of LLMs, researchers pay increasing attention to exploring their potential for sequential recommendation. Not only the extensive world knowledge stored in LLMs could serve as a rich source of background information for items [36], but also the reasoning capabilities of LLMs are able to augment the next item prediction [46]. When integrating LLMs into recommendation (LLM4Rec), there are two prevalent categories [31, 50]:

- **LLM as the Recommender.** It involves training from scratch [28], tuning [2, 8, 13], prompting [9], and in-context learning [20, 32] an LLM on recommendation data to serve as a recommender. Although studies within this category have substantiated that LLMs can be imbued with recommendation capabilities, they neglect established yet effective recommendation models.

- **LLM as the Enhancer.** It augments traditional recommenders with LLM tokens or embeddings [18, 19, 54]. It typically utilizes LLMs as feature extractors or text generators, given their exceptional ability to integrate diverse sources and forms of information, such as item metadata. Nonetheless, the actual recommendation process is still done by conventional models, leaving the LLMs' reasoning skills untouched.

Different from the aforementioned studies, LLaRA investigates aligning traditional sequential recommendation models with LLMs. It not only capitalizes on the sequential behavioral patterns learned by the well-established recommender models, but also utilizes the reasoning ability and world knowledge embedded within LLMs. In contrast to its concurrent work [57], LLaRA introduces the curriculum tuning strategy to achieve this alignment, ensuring a more

(a) Curriculum Prompt Tuning for Hybrid Item Representation.
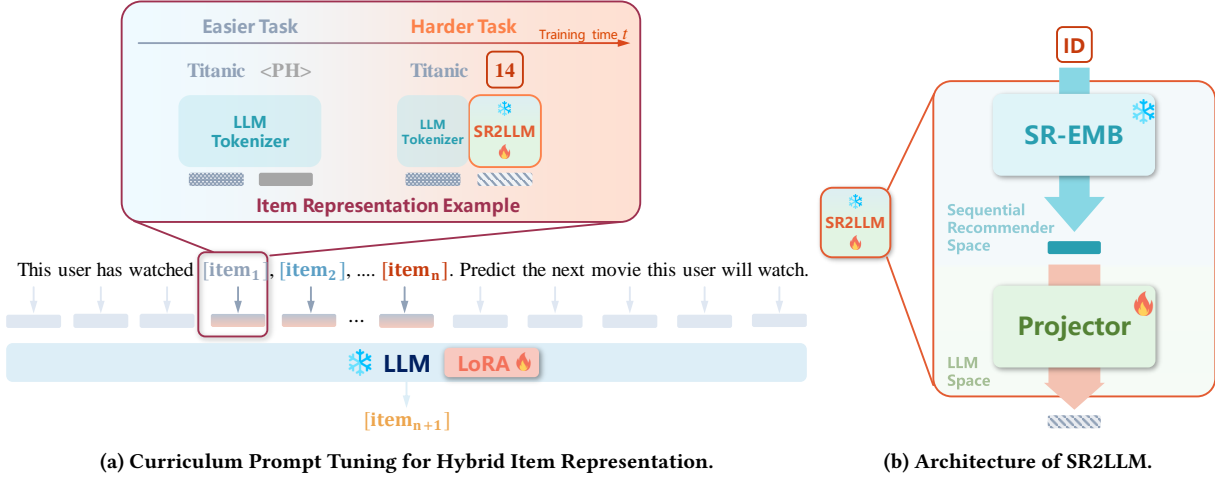
(b) Architecture of SR2LLM.

Figure 2: The LLaRA framework. (a) Sequential recommendation data is transformed into the instruction-tuning format. The item representation example illustrates the transition from pure textual tokens to the integration of the textual tokens with the behavioral token. (b) The sequential recommender is well-trained and frozen, while the trainable projector bridges the sequential recommender and LLM space.

stable learning procedure, and concentrates on list-wise ranking instead of the point-wise binary (yes/no) classification task.

3 PRELIMINARY

Task Formulation. Given a user who has chronologically engaged with item sequence $[i_1, i_2, \ldots, i_n]$, a sequential recommender entails predicting the next item i_{n+1} this user will interact with.

Curriculum Learning. Inspired by the pedagogical strategies in human education, curriculum learning [3] emphasizes training the model from simpler to more complex learning tasks. In general, it involves three critical stages [49]:

(1) Complexity Assessment: This initial stage quantifies the complexity of each data point or task, which is then used to assign a learning priority.

(2) Scheduler Formulation: Based on the complexity, a training scheduler is developed to arrange the sequence and frequency of tasks presented to the model, typically commencing with easier tasks and gradually advancing to harder ones.

(3) Training Execution: The curriculum learning process is implemented adhering to the predetermined progression.

Instruction Tuning. Instruction tuning emerges as a pivotal approach that can substantially boost LLMs to follow human task-specific instructions [37]. Specifically, it first reorganizes data into $\mathcal{Z} = \{(x_i, y_i)\}_{i=1,..,N}$, where x_i and y_i denote the textual instructions and the corresponding responses respectively. This pairing format not only encapsulates the task descriptions but also converts training data into a natural language format, thus creating a comprehensive instructional context. Subsequently, we can tune the LLMs with $\mathcal{Z}$ following the autoregressive objective [4, 45?]:

$$\max_{\Phi} \sum_{(x,y) \in \mathcal{Z}} \sum_{t=1}^{|y|} \log(P_\Phi(y_t|x, y_{<t})), \qquad (1)$$

where Φ is the parameters of the LLMs, with y_t referring to the t-th token of y, and $y_{<t}$ indicating the tokens preceding y_t.

Parameter Efficient Fine-Tuning. Fine-tuning all parameters of LLMs is time-consuming and resource-intensive. To alleviate this challenge, Parameter-Efficient Fine-Tuning (PEFT) [15, 26, 33] optimizes a smaller set of parameters, significantly reducing computational requirements while still achieving commendable performance. LoRA [21] is a typical PEFT algorithm, which keeps the LLM weights frozen and decomposes the updating weights into trainable low-rank matrices. The optimizing objective of LoRA can be formulated as follows:

$$\max_{\Theta} \sum_{(x,y) \in \mathcal{Z}} \sum_{t=1}^{|y|} \log \left(P_{\Phi_0 + \Delta\Phi(\Theta)}(y_t|x, y_{<t}) \right), \qquad (2)$$

where LoRA introduces parameters Θ, which are smaller in size in comparison to the original LLM parameters Φ_0.

4 LARGE LANGUAGE-RECOMMENDATION ASSISTANT (LLARA)

To incorporate the behavioral patterns learned by traditional recommenders into LLMs, we propose an end-to-end framework, Large Language-Recommendation Assistant (LLaRA), as depicted in Figure 2a. Specifically, beyond the text-only prompting, it exploits a hybrid prompting to align the behavioral representations, as derived from recommendation systems, with the language space of LLMs. It then employs curriculum learning — first focusing on text-only prompting, then progressively transitioning to hybrid prompting. This progressive strategy enables the LLM to familiarize the recommendation mechanism and internalize the behavioral knowledge of conventional recommenders. We now delve into the detailed architecture and training paradigm of LLaRA.

(a) Text-only prompting method.	(b) Hybrid prompting method.
Input: This user has watched Titanic [PH], Roman Holiday [PH], ... Gone with the wind [PH] in the previous. Please predict the next movie this user will watch. The movie title candidates are The Wizard of Oz [PH], Braveheart [PH],…, Waterloo Bridge [PH],… Batman & Robin [PH]. Choose only one movie from the candidates. The answer is	**Input:** This user has watched Titanic $[emb_s^{14}]$, Roman Holiday $[emb_s^{20}]$, ... Gone with the wind $[emb_s^{37}]$ in the previous. Please predict the next movie this user will watch. The movie title candidates are The Wizard of Oz $[emb_s^{5}]$, Braveheart $[emb_s^{42}]$,…, Waterloo Bridge $[emb_s^{20}]$,… Batman & Robin $[emb_s^{19}]$. Choose only one movie from the candidates. The answer is
Output: Waterloo Bridge.	**Output:** Waterloo Bridge.

Figure 3: Illustration of text-only and hybrid prompting method. (a) Text-only prompting represents items with the combination of the textual token and a placeholder token. (b) Hybrid prompting represents items with the integration of the textual token and the behavioral token. Note that <PH> indicates a special placeholder token, reserved for substitution by the behavioral token $<emb_s^i>$ throughout the progressive learning procedure.

4.1 Item Representation

Textual Token Representation. Textual features of items, such as titles and descriptions, are the key to harnessing the commonsense knowledge inherent in LLMs. Formally, for an item i with the text metadata txt_i, we obtain its textual tokens $<\textbf{emb}_t^i>$ as follows:

$$<\textbf{emb}_t^i> = \textbf{LLM-TKZ}(txt_i), \quad (3)$$

where **LLM-TKZ**$(\cdot)$ presents the LLM tokenizer and word embedding layer, encapsulating the process of transforming textual metadata into token representations. Such textual token representations of items, residing within the language space, are inherently compatible with LLMs.

Behavioral Token Representation. In parallel, conventional sequential recommender models, such as GRU4Rec [16], Caser [44], and SASRec [25], effectively capture sequential patterns within ID-based item embeddings after training on the historical interaction data. Formally, for item i, its ID-based representation learned by the conventional recommendation model is expressed as:

$$\textbf{e}_s^i = \textbf{SR-EMB}(i; \Theta_e), \quad (4)$$

where **SR-EMB**$(\cdot)$ is the function that generates the item embedding with the sequential recommender **SR** parameterized by Θ_e, and $\textbf{e}_s^i \in \mathbb{R}^d$ is the d-dimensional representation of item i.

In contrast to the item-aware texts that can be naturally inserted into the prompt and easily interpreted by LLMs, the ID-based item representations might be incompatible with the text nature of LLM prompts. Consequently, we view the ID-based representations as a distinct modality, separate from textual data. To bridge the modality gap, it is essential to map the ID-based representation space of recommenders into the language space of LLMs. This alignment allows LLMs to interpret and leverage the behavioral knowledge distilled by conventional recommenders.

To facilitate the alignment, we introduce a specialized module, SR2LLM, as illustrated in Figure 2b. Specifically, we project the ID-based item representation $\textbf{e}_s^i$ into the LLM space with a trainable projector **Proj** (*i.e.*, two-layer perceptions). This process results in the generation of a behavioral token representation, $<\textbf{emb}_s^i>$, formalized as:

$$<\textbf{emb}_s^i> = \textbf{Proj}(\textbf{e}_s^i; \Theta_p), \quad (5)$$

with Θ_p as the parameters of the trainable projector.

Hybrid Token Representation. Upon acquiring the textual tokens $<\textbf{emb}_t^i>$ and the behavioral token $<\textbf{emb}_s^i>$ for item i, we proceed to integrate these two components. This integration facilitates

a comprehensive description of item i, effectively combining the distinct yet complementary aspects captured by each token:

$$<\textbf{emb}_c^i> = \textbf{Concat}(<\textbf{emb}_t^i>, <\textbf{emb}_s^i>). \quad (6)$$

4.2 Hybrid Prompt Design

Text-Only Prompting. For converting sequential interaction data into training data suitable for LLM instruction tuning, our initial approach adopts a straightforward method known as text-only prompting. This approach represents items via textual metadata within the prompts, as illustrated in Figure 3a. The input prompts x encompass several key elements:

   (1) Task Definition: a clear description of the sequential recommendation task (*e.g.,* "predict the next movie this user will watch").

   (2) Interaction Sequence: the sequence of historical user-item interactions (*e.g.,* "Titanic <PH>, Roman Holiday <PH>, ..., Gone with the wind <PH>").

   (3) Candidate Set: the set of candidate items, from which the LLM is to generate responses to the given task (*e.g.,* "The Wizard of Oz <PH>, Braveheart <PH>, ..., Waterloo Bridge <PH>, ..., Batman & Robin <PH>").

   Within the input prompt, each item is represented using the textual tokens followed by a placeholder token. Additionally, the output y comprises the textual tokens corresponding to the next item with which the user will engage (*e.g.,* "Waterloo Bridge").

Hybrid Prompting. To incorporate behavioral insights captured by recommender models into the prompts, we devise a hybrid prompting method, as exhibited in Figure 3b. Formally, we consider a user u with a historical sequence of interactions involving items denoted as $h_1, h_2, \ldots, h_n$. The user is presented with a set of candidate items, represented as $\mathbb{C} = \{c_1, c_2, \ldots, c_m\}$, from which the user may select the next item of interest. Thus the three primary components of hybrid input prompts x are transformed correspondingly as follows:

(1) **Task Definition:** identical to the text-only prompting method, which describes the sequential recommendation task in text.
(2) **Interaction Sequence with Hybrid Item Representations:** the sequence of historical user-item interactions, represented as $<\textbf{emb}_c^{h_1}>, <\textbf{emb}_c^{h_2}>, \ldots, <\textbf{emb}_c^{h_n}>$ (*e.g.,* Titanic $<\textbf{emb}_s^{14}>$, Roman Holiday $<\textbf{emb}_s^{20}>$, ..., Gone with the wind $<\textbf{emb}_s^{37}>$).
(3) **Candidate Set with Hybrid Item Representations:** the set of item candidates represented with the integration of textual and behavioral tokens as $<\textbf{emb}_c^{c_1}>, <\textbf{emb}_c^{c_2}>, \ldots, <\textbf{emb}_c^{c_m}>$, from

which the LLM is expected to generate responses (*e.g.,* The Wizard of Oz <$\mathbf{emb}_s^5$>, Braveheart <$\mathbf{emb}_s^{42}$>,..., Waterloo Bridge <$\mathbf{emb}_s^{20}$>,..., Batman & Robin <$\mathbf{emb}_s^{19}$>).

This approach utilizes a fusion of textual and behavioral tokens, as formulated in Equation (6), to represent items. This contrasts with the text-only prompts, which rely solely on textual tokens as outlined in Equation (3), thereby enriching the prompt with a more comprehensive understanding of user-item interactions.

Our hybrid prompt design facilitates integration of textual metadata and ID-based item embeddings sourced from a well-trained recommender model. This design addresses the limitations of prompts that rely exclusively on either ID-based or textual data, thereby generating more accurate recommendations.

4.3 Curriculum Prompt Tuning

Considering the design of LLMs, which predominantly train on data in text, the task of comprehending modalities — behavioral tokens distilled from recommender models — presents a notable challenge. While the text-only prompting aligns closely with the LLMs' training and is thus more readily assimilated, the hybrid prompting, representing a deviation from typical language data, introduces a more complex task.

Drawn inspiration from curriculum learning [3], which emphasizes the importance of training the model from simple to more challenging learning tasks, we design a curriculum prompt tuning scheme in LLaRA. In general, the tuning process begins by focusing on the more straightforward prompting method — text-only prompting method. This initial phase allows the model to establish a fundamental grasp of the sequential recommendation task. Subsequently, we gradually introduce the hybrid prompting method that incorporates behavioral tokens, thereby elevating the complexity of the tuning process. This step-wise strategy ensures that the model is not overwhelmed by the complex task. Ultimately, our LLM-based recommender will be fully integrated with the hybrid item representation. This entire learning trajectory is shown as the gradient-colored rectangle in Figure 2a.

Formally, this learning process can be articulated through the subsequent stages, corresponding point-to-point with the three pivotal phases of curriculum learning.

(1) Complexity Assessment: The initial step of curriculum learning is to assess the complexity of each task. In LLaRA, the task complexity is highly related to the integration of behavioral tokens in the hybrid prompt design. Therefore, we define the easy and hard learning tasks, where the easy task adopts the sequential data reformatted into the text-only prompts as depicted in Figure 3a, whereas the hard task employs the data reformatted into the hybrid prompts as elucidated in Figure 3b. Specifically, the loss function of the easy task can be formulated as:

$$L_{easy}(x^e, y^e) = -\sum_{t=1}^{|y^e|} \log\left(P_{\Phi_0+\Delta\Phi(\Theta)}(y_t^e|x^e, y_{<t}^e)\right), \quad (7)$$

where (x^e, y^e) is the text-only prompts shown in Figure 3a. Besides, the loss function of the hard counterpart can be formulated as:

$$L_{hard}(x^h, y^h) = -\sum_{t=1}^{|y^h|} \log\left(P_{\Phi_0+\Delta\Phi(\Theta)+\Theta_p+\Theta_e}(y_t^h|x^h, y_{<t}^h)\right), \quad (8)$$

where Θ_p and Θ_e are the parameters of the projector and the embedding layer of the conventional sequential recommender, respectively, and (x^h, y^h) represents the hybrid prompt in Figure 3b.

(2) Scheduler Formulation: After acquiring the learning objectives of the easy and hard tasks in Equation (7) and (8), respectively, we can formulate the curriculum scheduler by transferring from the easy task to the hard task gradually in the training process. Specifically, we denote $p(\tau)$ as the probability of learning the hard task at training time τ, while $1 - p(\tau)$ is the probability of the easy task, correspondingly. Naturally, p should be small at the beginning and gradually increase in the learning process, which can be formulated in a continuous manner:

$$p(\tau) = \frac{\tau}{T} \quad (0 \le \tau \le T), \quad (9)$$

with the total training time accumulated to T.

(3) Training Execution: To strike a balance between efficiency and efficacy, we conduct LoRA tuning as introduced in Equation (2) for the LLM, while training the projector at the same time. Formally, we define the indicator function:

$$\mathbb{I}(\tau) = \begin{cases} 1, & \text{learning hard task } (w.p.\ p(\tau)) \\ 0, & \text{learning easy task } (w.p.\ 1 - p(\tau)) \end{cases}. \quad (10)$$

Therefore, the learning objective of LLaRA evolves from the easier task to the harder task:

$$\min_{\Theta,\Theta_p} \sum_{(x,y)\in\mathcal{Z}} \left((1 - \mathbb{I}(\tau))\ L_{easy}(x, y) + \mathbb{I}(\tau)\ L_{hard}(x, y)\right). \quad (11)$$

This gradual learning process effectively facilitates the injection of an additional modality, thereby actualizing the hybrid prompting method. By adopting the curriculum prompt tuning strategy, we ensure a seamless transition from the model's initial understanding of textual metadata to its eventual comprehension of more complex ID-based item embeddings from traditional recommenders. This strategy not only acquaints LLMs with the recommendation mechanism, but also enhances LLMs with the behavioral knowledge encapsulated in the sequential recommenders.

5 EXPERIMENTS AND RESULTS

In this section, we evaluate our proposed framework LLaRA on three real-world datasets, and compare it with several baselines, including traditional sequential recommender models and LLM4Rec models. Additionally, we carry out two ablation studies to demonstrate the substantial enhancements brought about by the hybrid prompting method and curriculum prompt tuning strategy of LLaRA. Furthermore, we present case studies to explicitly show our advantages over baselines. To validate the superiority of our framework, we will showcase it by answering research questions as follows.

- **RQ1:** How does LLaRA perform compared with traditional sequential recommender models and LLM-based methods?

Table 1: Statistics of Datasets.

Dataset	MovieLens	Steam	LastFM
# Sequence	943	11,938	1,220
# Item	1,682	3,581	4,606
# Interaction	100,000	274,726	73,510

- **RQ2:** How does our hybrid prompting perform in comparison to other forms of item representation in prompt design?
- **RQ3:** How does our curriculum learning scheme measure against other modality injection methods?

5.1 Experimental Settings

5.1.1 Datasets.

- **MovieLens** [14] is a commonly-used movie recommendation dataset that contains user ratings and movie titles.
- **Steam** [25] encompasses user reviews for video games on the Steam Store, in addition to game titles.
- **LastFM** [5], collected from the Last.fm online music platform, includes user-artist listening relationships and the names of artists.

Given that tuning LLMs is more time-consuming than training traditional recommenders, we choose the MovieLens100K dataset for our experiment to ensure that the dataset size remains manageable. Regarding the Steam dataset, we initially eliminate users with fewer than 20 reviews, aligning with the processing method employed for MovieLens. Then, we randomly select a third of the users and a third of the games, maintaining their interactions to derive a dataset of a moderate size. For all three datasets, we arrange sequences chronologically and divide the data into train, validation, and test subsets at a ratio of 8:1:1. This partitioning approach guarantees that subsequent interactions do not appear in the training data, thereby circumventing any potential information leakage [24]. Detailed statistics of the datasets are provided in Table 1. Moreover, we retain the last 10 interactions as the historical sequence, padding sequences with fewer than 10 interactions.

5.1.2 Implementation Details.
We select Llama2-7B [46] as the LLM backbone. To ensure the flexibility of our textual interface, the instruction format for training and testing is randomly sampled from several prompts. Our implementations for conventional recommenders follow [52], employing the Adam optimizer, with a learning rate of 0.001, an embedding dimension d of 64, and a batch size of 256. Furthermore, we conduct a grid search in [1e-3, 1e-4, 1e-5, 1e-6, 1e-7] for the coefficient of L2 regularization. To mitigate the impact of randomness, we report the average outcomes of five runs using different random seeds. For all methods related to LLMs, each experiment is trained for a maximum of 5 epochs, with a batch size of 128. We employ a warm-up strategy for the learning rate, initiated with 1/100 of the maximum learning rate, and adjust it over steps using a cosine scheduler.

5.1.3 Evaluation Metrics.
For each sequence, we randomly select 20 non-interacted items to construct the candidate set, ensuring the inclusion of the correct subsequent item. LLaRA and other baseline models aim to identify the correct item from this candidate set, and their performance is evaluated using the HitRatio@1 metric. With appropriate prompting, LLM-based recommenders can generate a single candidate item as required. As for traditional models, we
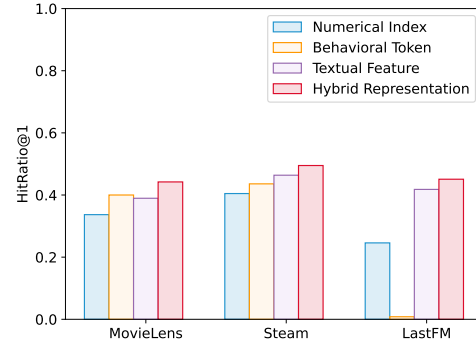

Figure 4: The performance comparison of different item representation methods (*i.e.,* numerical index, behavioral token, textual feature, and hybrid representation). The hybrid representation is adopted in LLaRA.

select the candidate item with the highest probability as the prediction. Meanwhile, since LLaRA employs a generative paradigm for prediction, which may yield invalid responses such as nonsensical words or items outside the candidate sets, we introduce an additional metric — valid ratio. It quantifies the proportion of valid responses (*i.e.,* items in the candidate set) across all sequences, serving as a measure of the models' capability of instruction following.

5.2 Performance Comparison (RQ1)

In this section, we compare LLaRA against both traditional and LLM-based baselines, taking into account metrics of both HitRatio@1 and valid ratio on MovieLens, Steam, and LastFM datasets, to showcase the effectiveness and robustness of LLaRA.

5.2.1 Baselines.

- Traditional Sequential Recommenders: **GRU4Rec** [16], **Caser** [44], and **SASRec** [25], are RNN-based, CNN-based, and attention-based sequential recommenders, respectively.
- LLM-based Models: (1) **Llama2** [46] is a well-known open-source LLM released by Meta. (2) **GPT-4** [36], released by OpenAI, is a milestone of LLMs excelling in various tasks. (3) **MoRec** [1] [54] enhances the traditional recommenders by encoding item's modality features, such as text features. (4) **TALLRec** [2] [2] conducts instruction tuning for LLMs on recommendation corpus.

5.2.2 Results.
We implement LLaRA framework on item embeddings derived from three traditional sequential recommendation baselines (*i.e.,* GRU4Rec, Caser, and SASRec). Comparing LLaRA with the aforementioned baseline models, the results are shown in Table 2 [3]. The observations can be summarized as follows.

(a) LLaRA outperforms all baselines on all three datasets. Specifically, it achieves the highest HitRatio@1 metric of 0.4737, 0.4949 and 0.4508 on MovieLens, Steam and LastFM, respectively. This

---

[1]For MoRec, we adopt BERT as the text encoder and SASRec as the recommender backbone, consistent with the officially provided implementation.

[2]TALLRec predicts YES/NO for the target item, thus we adapt it to our setting – selecting the next item from a provided candidate set.

[3]Note that the relative improvement of GRU, Caser, and SASRec is calculated by the LLaRA implemented on the item embeddings derived from their corresponding models, while that of LLM-based methods is calculated by the LLaRA implemented on SASRec.

Jiayi Liao et al.

Table 2: The Results of LLaRA compared with traditional sequential recommender models and LLMs-based methods. Bold and underlined indicate the best and the second-best performance, respectively. *(p-value $<< 0.05$).

		MovieLens*		Steam*		LastFM	
		ValidRatio	HitRatio@1	ValidRatio	HitRatio@1	ValidRatio	HitRatio@1
Traditional	GRU4Rec	1.0000	0.3750	1.0000	0.4168	1.0000	0.2616
	Caser	1.0000	0.3861	1.0000	0.4368	1.0000	0.2233
	SASRec	1.0000	0.3444	1.0000	0.4010	1.0000	0.2233
LLM-based	Llama2	0.4421	0.0421	0.1653	0.0135	0.3443	0.0246
	GPT-4	0.9895	0.2000	0.9798	0.3626	1.0000	0.3770
	MoRec	1.0000	0.2822	1.0000	0.3911	1.0000	0.1652
	TALLRec	0.9263	0.3895	0.9840	0.4637	0.9836	0.4180
Ours	LLaRA (GRU4Rec)	0.9684	<u>0.4421</u>	0.9975	<u>0.4924</u>	0.9836	<u>0.4344</u>
	LLaRA (Caser)	0.9684	**0.4737**	0.9966	0.4874	0.9918	<u>0.4344</u>
	LLaRA (SASRec)	0.9684	<u>0.4421</u>	0.9975	**0.4949**	1.0000	**0.4508**

validates its effective integration of traditional sequential information with the extensive world knowledge and robust reasoning capabilities of LLMs.

(b) As for traditional sequential recommenders (*i.e.,* GRU4Rec, Caser, and SASRec), their HitRatio@1 scores are lower than those of LLaRA. These models make predictions solely based on the behavioral patterns of users, without integrating any semantic information about items. This highlights the importance of incorporating world knowledge about items into the recommendation process.

(c) When it comes to LLM-based methods, we can analyze them from two perspectives. Firstly, the relatively poor performance of vanilla LLMs (*i.e.,* Llama2 and GPT-4) suggests that adapting LLMs to recommendation tasks is crucial for enhancing their performance in this domain. Secondly, the LLM4Rec methods (*i.e.,* MoRec and TALLRec) show some improvements over the standalone LLM methods. However, their recommendation ability, as denoted by the HitRatio@1 metric, is still lower than that of LLaRA. MoRec overlooks the reasoning ability of LLMs, while TALLRec neglects to incorporate traditional sequential recommenders. This highlights the need for a more comprehensive approach that combines the strengths of both LLMs and traditional recommendation models.

(d) LLaRA achieves a high validity ratio of over 95% on all datasets, illustrating the model's instruction-following abilities when generating recommendations. It's worth noting that all generative methods that incorporate LLMs might generate invalid answers. For instance, Llama2, which serves as the backbone LLM of LLaRA, only achieves valid ratios of 0.4421, 0.1653, and 0.3443 on the MovieLens, Steam, and LastFM datasets, respectively. Remarkably, LLaRA's significant improvement in valid ratios can be attributed to the fact that LLaRA has been instruction-tuned on the sequential recommendation task.

5.3 Impact of Hybrid Item Representation (RQ2)

We conduct experiments to evaluate the item representation methods in sequential recommendation.

- **Numerical Index:** The items in the textual prompts are represented as numerical indices.
- **Behavioral Token:** The items are represented using behavioral tokens projected from the sequential recommender space, employing the identical projector architecture as LLaRA.

Table 3: The HitRatio@1 of LLaRA compared with other learning strategies. CL denotes curriculum learning and bold indicates the best performance.

	MovieLens	Steam	LastFM
Direct	0.4211	0.4899	**0.4508**
Two-stage	0.4316	0.4840	0.4344
LLaRA (CL)	**0.4421**	**0.4949**	**0.4508**

- **Textual Feature:** The items in the textual prompts are represented by their respective titles.
- **Hybrid Representation:** LLaRA proposes to represent items with the fusion of behavioral tokens and textual tokens.

The results are shown in Figure 4, and we can observe that the item representation approach utilized by LLaRA surpasses other methods in terms of HitRatio@1 across all three datasets. This not only corroborates the effectiveness of our innovative item representation method, but also illustrates the insufficiency of solely relying on semantic information (*i.e.,* textual metadata) or sequential information (*i.e.,* behavioral tokens).

Concerning numerical indices, no information is initially stored in LLMs for these indices. The numerical indices are processed as plain text by LLMs, culminating in their separation into several tokens by the LLM tokenizer. In the case of behavioral tokens, the LLM merely capitalizes on the distribution of the inputted behavioral embeddings, without eliciting the knowledge encapsulated within the LLM. As for textual features, users' behavioral patterns are absent, allowing the LLM to solely infer the correlations among items in a user's historical interactions, guided by the background knowledge of these items preserved in the LLM. In contrast, LLaRA integrates both world knowledge and sequential information, thereby improving performance in sequential recommendation.

5.4 Impact of Curriculum Prompt Tuning (RQ3)

This section delves into the development of an optimal learning strategy for modality integration, by comparing three schemes:

(1) **Direct Training:** The hybrid item representation is employed consistently during training.

(2) **Two-Stage Training:** The training process is split into two stages. Initially, Llama2 is fine-tuned on the easy task wherein the item representation is solely comprised of item titles.
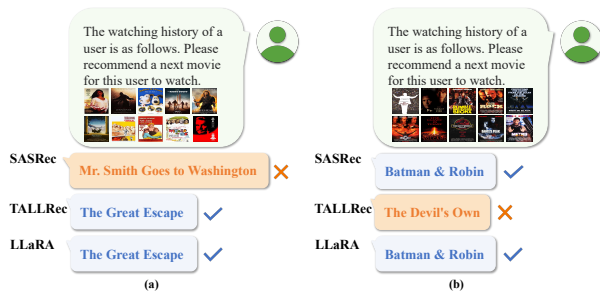
Figure 5: Case studies. (a) The user prefers adventure and war genres according to the viewing history. With the world knowledge about these movies, TALLRec and LLaRA correctly recommend "The Great Escape". (b) SASRec and LLaRA recommend "Batman & Robin", according to the sequential behavioral patterns of users.

(3) LLaRA (CL): LLaRA framework adopts a single-stage curriculum learning approach. Our curriculum learning strategy instructs the model to transition gradually from the basic text-only prompting to the hybrid prompting.

All training procedures encompass a total of five epochs for fair, while in the case of the two-stage method, the epoch number for the first and second stages is 2 and 3, respectively.

A careful analysis of the results, presented in Table 3, reveals that curriculum learning employed by LLaRA, consistently outperforms the other baseline methods across all datasets. Specifically, the direct training method confounds the model with the hard task throughout the entire process, while the two-stage training approach fine-tunes Llama2 on the text-only and hybrid prompts in the first and second stages, respectively. LLaRA starts from the easy task and progressively changes to the hard task utilizing a sampler to schedule the training process. The improvement brought by this gradual learning method underscores the effectiveness of our curriculum prompt tuning scheme.

5.5 Case Studies

We select two typical cases to analyze the impact of the world knowledge within LLMs, pertaining to items, as well as the behavioral patterns exhibited by users on the sequential recommendation task. To illustrate these two factors, we choose the answers generated by three models, SASRec, TALLRec, and LLaRA.

5.5.1 World Knowledge in LLMs. For a user who sequentially watched "Ruby in Paradise", "The Shawshank Redemption", "Wallace & Gromit: The Best of Aardman Animation", "The Right Stuff", "Braveheart", "The Princess Bride", "North by Northwest", "Some Like It Hot", "The Wizard of Oz", and "The Hunt for Red October", SASRec predicted the next film to be "Mr. Smith Goes to Washington", while TALLRec and LLaRA recommended "The Great Escape". The user's actual subsequent interaction was indeed "The Great Escape" as shown in Figure 5a.

We can observe that, the world knowledge about movies inherent in the LLM can be highly beneficial for the sequential recommendation, as demonstrated here. The genres of films this user has watched include adventure ("The Princess Bride", "The Wizard of Oz", "North by Northwest") and war ("Braveheart", "The Hunt for

Red October"). Since the LLM was capable of analyzing this user's watching history and understanding that the user has a preference for adventure and war genres, this insight allowed the LLM to correctly predict that the user would choose "The Great Escape" (a war adventure film) rather than "Mr. Smith Goes to Washington" (a political drama). LLaRA, benefiting from the integration of the LLM's world knowledge, also forecasted the correct choice.

5.5.2 Sequential Behavioral Patterns in Traditional Sequential Recommenders. A user sequentially watched the following ten films: "Mr. Holland's Opus", "Courage Under Fire", "Rumble in the Bronx", "The Rock", "Men in Black", "Con Air", "Volcano", "The Lost World: Jurassic Park", "Dante's Peak", and "Metro" as shown in the Figure 5b. TALLRec predicted the subsequent film to be "The Devil's Own"; whereas both SASRec and LLaRA recommended "Batman & Robin", which aligns with the user's actual interaction.

TALLRec, based on background knowledge, may have inferred that the user prefers action, adventure, or thriller films over superhero movies. "The Devil's Own" is an action thriller, while "Batman & Robin" is a superhero film. However, SASRec, by analyzing the user's interaction history, unearthed sequential behavioral patterns and recommended the correct film. LLaRA, due to the incorporation of information from SASRec, also predicted the correct answer. This case illustrates that the sequential behavioral patterns of users hold substantial importance in sequential recommendation.

6 CONCLUSION AND DISCUSSION

In this paper, we introduce a novel framework, Large Language-Recommendation Assistant (LLaRA) that integrates traditional recommender models with LLMs, and transforms the sequential recommendation task into language modeling. In particular, LLaRA adopts curriculum learning that gradually injects sequential patterns learned by traditional sequential recommenders into the tuning process of LLMs. Empirical results show that LLaRA outperforms all baseline models in sequential recommendation, demonstrating its effectiveness and promising performance. Ablation studies underscore the essential role of both the hybrid prompting method and the curriculum prompt tuning strategy.

This work marks an initial step in transitioning from the traditional recommender models to a more sophisticated approach underpinned by LLMs and opens up new research possibilities. It lays the groundwork by proposing an alignment mechanism to bridge conventional recommender models with LLMs. In the future, researchers could continue to explore a unified recommendation framework, with natural language as the interface, for more complex and diverse recommendation scenarios. We hope the development of LLaRA paves the way for a new era of personalized, integrated, and universal recommender systems.

REFERENCES

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. In *NeurIPS*.

[2] 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 *RecSys*. ACM, 1007–1014.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML (ACM International Conference Proceeding Series, Vol. 382)*. ACM, 41–48.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *NeurIPS*.

[5] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In *Proceedings of the 5th ACM conference on Recommender systems* (Chicago, IL, USA) *(RecSys 2011)*. ACM, New York, NY, USA.

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

[7] Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. ChatLaw: Open-Source Legal Large Language Model with Integrated External Knowledge Bases. arXiv:2306.16092 [cs.CL]

[8] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems. *CoRR* abs/2205.08084 (2022).

[9] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. In *RecSys*. ACM, 1126–1132.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.

[11] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. PaLM-E: An Embodied Multimodal Language Model. In *ICML (Proceedings of Machine Learning Research, Vol. 202)*. PMLR, 8469–8488.

[12] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations. *ACM Trans. Inf. Syst.* 39, 1 (2020), 10:1–10:42.

[13] 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 *RecSys*. ACM, 299–315.

[14] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19.

[15] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a Unified View of Parameter-Efficient Transfer Learning. In *International Conference on Learning Representations*.

[16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR (Poster)*.

[17] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 2023. 3D-LLM: Injecting the 3D World into Large Language Models. *arXiv preprint arXiv:2307.12981* (2023).

[18] Yupeng Hou, Zhankui He, Julian J. McAuley, and Wayne Xin Zhao. 2023. Learning Vector-Quantized Item Representation for Transferable Sequential Recommenders. In *WWW*. ACM, 1162–1171.

[19] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *KDD*. ACM, 585–593.

[20] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian J. McAuley, and Wayne Xin Zhao. 2023. Large Language Models are Zero-Shot Rankers for Recommender Systems. *CoRR* abs/2305.08845 (2023).

[21] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*. OpenReview.net.

[22] Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2023. How to Index Item IDs for Recommendation Foundation Models. *CoRR* abs/2305.06569 (2023).

[23] Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, Yi Ren, Zhou Zhao, and Shinji Watanabe. 2023. AudioGPT: Understanding and Generating Speech, Music, Sound, and Talking Head. *CoRR* abs/2304.12995 (2023).

[24] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2023. A Critical Study on Data Leakage in Recommender System Offline Evaluation. *ACM Trans. Inf. Syst.* 41, 3 (2023), 75:1–75:27.

[25] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. IEEE Computer Society, 197–206.

[26] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *EMNLP (1)*. Association for Computational Linguistics, 3045–3059.

[27] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. *CoRR* abs/2301.12597 (2023).

[28] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian J. McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. In *KDD*. ACM, 1258–1267.

[29] Sihang Li, Zhiyuan Liu, Yanchen Luo, Xiang Wang, Xiangnan He, Kenji Kawaguchi, Tat-Seng Chua, and Qi Tian. 2024. Towards 3D Molecule-Text Interpretation in Language Models. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=xI4yNlkaqh

[30] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. 2022. Supervision Exists Everywhere: A Data Efficient Contrastive Language-Image Pre-training Paradigm. In *ICLR*. OpenReview.Net.

[31] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2023. How Can Recommender Systems Benefit from Large Language Models: A Survey. *CoRR* abs/2306.05817 (2023).

[32] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. *CoRR* abs/2304.10149 (2023).

[33] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT Understands, Too. *CoRR* abs/2103.10385 (2021).

[34] Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. 2023. MolCA: Molecular Graph-Language Modeling with Cross-Modal Projector and Uni-Modal Adapter. In *EMNLP*. Association for Computational Linguistics, 15623–15638.

[35] Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. 2023. Macaw-LLM: Multi-Modal Language Modeling with Image, Audio, Video, and Text Integration. *arXiv* (2023).

[36] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023).

[37] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

[38] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *RecSys*. ACM, 130–137.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 8748–8763.

[40] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

[41] Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large Language Models Encode Clinical Knowledge. arXiv:2212.13138 [cs.CL]

[42] 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 *CIKM*. ACM, 1441–1450.

[43] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *DLRS@RecSys*. ACM, 17–22.

[44] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*. ACM, 565–573.

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal

Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023).

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR* abs/2307.09288 (2023).

[47] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal Few-Shot Learning with Frozen Language Models. In *NeurIPS*. 200–212.

[48] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet A. Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. In *IJCAI*. ijcai.org, 6332–6338.

[49] Xin Wang, Yudong Chen, and Wenwu Zhu. 2022. A Survey on Curriculum Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 9 (2022), 4555–4576.

[50] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2023.

[51] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David S. Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. *CoRR* abs/2303.17564 (2023).

[52] Zhengyi Yang, Xiangnan He, Jizhi Zhang, Jiancan Wu, Xin Xin, Jiawei Chen, and Xiang Wang. 2023. A Generic Learning Framework for Sequential Recommendation with Distribution Shifts. In *SIGIR*. ACM, 331–340.

[53] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *WSDM*. ACM, 582–590.

[54] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to Go Next for Recommender Systems? ID- vs. Modality-based Recommender Models Revisited. In *SIGIR*. ACM, 2639–2649.

[55] An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024. On Generative Agents in Recommendation. In *SIGIR*.

[56] Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding. *arXiv* (2023).

[57] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. CoLLM: Integrating Collaborative Embeddings into Large Language Models for Recommendation. *CoRR* abs/2310.19488 (2023).

[58] Yuyue Zhao, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten de Rijke. 2024. Let Me Do It For You: Towards LLM Empowered Recommendation via Tool Learning. In *SIGIR*.

[59] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* (2023).

A Survey on Large Language Models for Recommendation. *CoRR* abs/2305.19860 (2023).