TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation

KEQIN BAO*, University of Science and Technology of China, China
JIZHI ZHANG*, University of Science and Technology of China, China
YANG ZHANG, University of Science and Technology of China, China
WENJIE WANG, National University of Singapore, Singapore
FULI FENG, University of Science and Technology of China, China
XIANGNAN HE, University of Science and Technology of China, China

Large Language Models (LLMs) have demonstrated remarkable performance across diverse domains, thereby prompting researchers to explore their potential for use in recommendation systems. Initial attempts have leveraged the exceptional capabilities of LLMs, such as rich knowledge and strong generalization through In-context Learning, which involves phrasing the recommendation task as prompts. Nevertheless, the performance of LLMs in recommendation tasks remains suboptimal due to a substantial disparity between the training tasks for LLMs and recommendation tasks, as well as inadequate recommendation data during pre-training. To bridge the gap, we consider building a *Large Recommendation Language Model* by tunning LLMs with recommendation data. To this end, we propose an efficient and effective Tuning framework for Aligning LLMs with Recommendation, namely TALLRec. We have demonstrated that the proposed TALLRec framework can significantly enhance the recommendation capabilities of LLMs in the movie and book domains, even with a limited dataset of fewer than 100 samples. Additionally, the proposed framework is highly efficient and can be executed on a single RTX 3090 with LLaMA-7B. Furthermore, the fine-tuned LLM exhibits robust cross-domain generalization. Our code and data are available at https://anonymous.4open.science/r/LLM4Rec-Recsys.

CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: Recommendation, Large Language Models, Instruction Tuning

ACM Reference Format:
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. 1, 1 (July 2023), 15 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Large Language Models (LLMs) have exhibited remarkable proficiency in generating text that closely resembles human language and in performing a wide range of tasks, including Natural Language Processing [3, 29, 54], Robotics [9, 42, 56], and Information Retrieval [22, 23, 47]. Prior research has also demonstrated the knowledge-rich and compositional

*The two authors contributed equally to this work and are listed alphabetically.
Authors' addresses: Keqin Bao*, baokq@mail.ustc.edu.cn, University of Science and Technology of China, China; Jizhi Zhang*, cdzhangjizhi@mail.ustc.edu.cn, University of Science and Technology of China, China; Yang Zhang, zy2015@mail.ustc.edu.cn, University of Science and Technology of China, China; Wenjie Wang, wenjiewang96@gmail.com, National University of Singapore, Singapore; Fuli Feng, fulifeng93@gmail.com, University of Science and Technology of China, China; Xiangnan He, xiangnanhe@gmail.com, University of Science and Technology of China, China.

Manuscript submitted to ACM

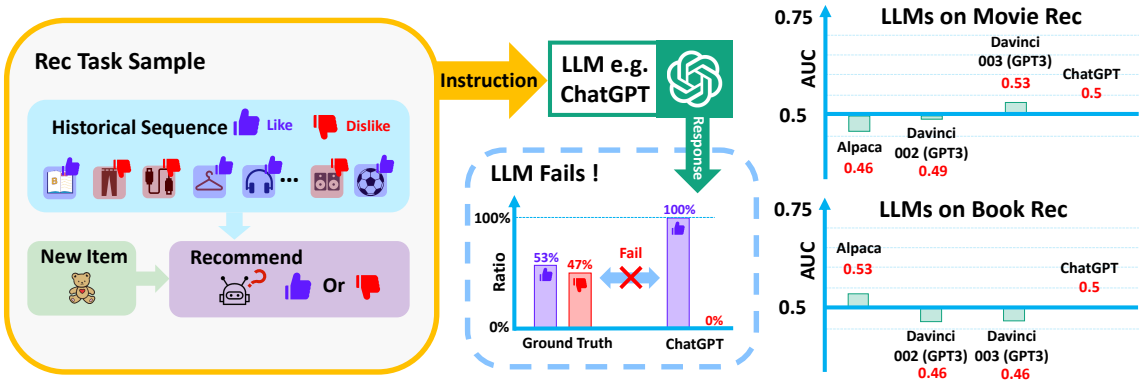

Fig. 1. This diagram shows how we test the ability of LLMs to make recommendations. LLMs are given the task of predicting whether a user like the next item based on their interaction history. We use In-context Learning with ChatGPT to solve this task but find that it consistently provides a single uniform answer or refused to answer, making it unsuitable for this recommendation task. In our experiment on Movie and Book data, we ignore samples that LLM refuse to answer and found that it performed no better than random guessing. For more information on how we construct our data, please refer to the experiment section.

generalization capabilities of LLMs [33, 40, 53]. Only given appropriate instructions, these models are able to learn how to solve unseen tasks and inspire their own knowledge to achieve a high level of performance [30]. The aforementioned capabilities of LLM present promising opportunities to address the current challenges requiring strong generalization and rich knowledge in the recommendation field. In this light, it is valuable to explore the integration of LLMs into recommender systems, which has received limited attention in prior research.

In recent initial attempts [11, 46], achieving the target relies on *In-context Learning* [2], which is typically implemented through the official OpenAI API [1]. They regard the LLM as a toolformer [41] of traditional recommendation models (such as MF [25] and LightGCN [14]), *i.e.,* the LLM is used for re-ranking the candidate items filtered by these models. However, these approaches only reach a comparable performance with traditional models [11, 46]. Worse still, using only In-context Learning may fail to make recommendations. As shown in Figure 1, we find that ChatGPT either refuses to answer or believes that users will like the new item. Therefore, it is critical to further explore an appropriate way for more effective leverage of LLMs in the recommendation.

We postulate that the failure of using only In-context Learning is because of two reasons: 1) LLMs may not align well with the recommendation task due to the huge gap between language processing tasks for training LLMs and recommendation. Besides, recommendation oriented corpus is very limited during the training phase of LLMs. 2) The effect of LLMs is restricted by the underlying recommendation models, which may fail to include target items in their candidate lists due to their limited capacity. Therefore, we consider building a *Large Recommendation Language Model* (LRLM) to bridge the gap between LLMs and the recommendation task and better stimulate the recommendation capabilities of LLMs in addition to In-context Learning.

Toward this goal, we focus on tuning LLMs with the recommendation task. Considering that instruction tuning is core to letting the LLM learn to solve different tasks and have strong generalization ability [20, 21, 36], we design a tuning procedure to facilitate the acquisition of the recommendation task by the LLM. Elaborately, we structure our training data in a manner akin to the instruction tuning process and subsequently train the LLM after the instruction tuning stage. Moreover, given that LLM training necessitates a substantial amount of data, we opt to employ a lightweight fine-tuning approach to efficiently adapt the LLMs to the recommendation task. To sum up, we propose an effective

and efficient tuning framework to align LLMs for making recommendations (TALLRec) with a small number of tuning samples and computational resource consumption.

Specifically, we apply the TALLRec framework on the LLaMA-7B model [45] with a LoRA [19] architecture, which ensures the framework can be deployed on a Nvidia RTX 3090 (24GB) GPU. We conduct detailed experiments in knowledge-rich recommendation scenarios of movies and books, where the tuned LLaMA-7B model outperforms traditional recommendation models and In-context Learning with GPT3.5, a much stronger LLM than LLaMA-7B. Furthermore, the results also validate the efficiency and robustness of our framework: 1) our TALLRec framework can quickly inspire the recommendation capability with just a few examples (less than 100); and 2) the model trained via our framework also has a strong generalization ability across different domains (*e.g., movie* $\rightarrow$ *book*).

In total, our contributions are summarized as follows:

- We study a new problem in recommendation — aligning the LLMs with recommendation, where we reveal the limitations of In-context Learning-based approaches and underscore the significance of instruction tuning.
- We introduce a new TALLRec framework to build Large Recommendation Language Models, which enables the effective and efficient tuning of LLMs for recommendation with low GPU cost and few tuning samples.
- We conduct extensive experiments, validating the effectiveness and efficiency of the proposed framework, and uncovering its exceptional robustness with seamless navigation across different domains.

2 TALLREC

In this section, we first introduce the preliminary knowledge for tuning LLMs and our task formulation, and then present the proposed TALLRec framework.

2.1 Preliminary

Instruction Tuning. Instruction tuning constitutes a crucial component of the LLM training process. It is a widely employed technique that facilitates the ability of LLMs by means of fine-tuning a diverse set of human-annotated instructions and responses [33]. This stage confers upon the model a robust generalization capability, thereby endowing it with the capacity to proficiently address unseen tasks and ensuring its adaptability to novel scenarios and challenges. In detail, instruction tuning has four general steps (referring to the example in Table 1):

Table 1. Example of tuning data for instruction tuning in a translation task.

Instruction Input	
Task Instruction:	Translate from English to Chinese.
Task Input:	Who am I ?
Instruction Output	
Task Output:	我是谁?

- Step 1: Identify a task and articulate an instruction using natural language to effectively solve the task. The instruction can encompass a clear definition of the task, as well as specific solutions to address it. These descriptions are referred to as an "*Task Instruction*" throughout this study.
- Step 2: Express the input and output of the task in natural language and define them as "*Task Input*" and "*Task Output*".
- Step 3: Integrate the "*Task Instruction*" and "*Task Input*" together to form the "*Instruction Input*", and take the "*Task Output*" as the corresponding "*Instruction Output*", for each data point.
- Step 4: Train the LLM based on the formatted "*Instruction Input*" and "*Instruction Output*" pair data.

Task Formulation. The primary aim of this study is to facilitate the alignment of LLMs with recommendation tasks, with the goal of unlocking their potential for recommendations. To achieve this objective, we conceptualize our task as the expeditious and effective training of a recommendation model, utilizing a restricted amount of user historical

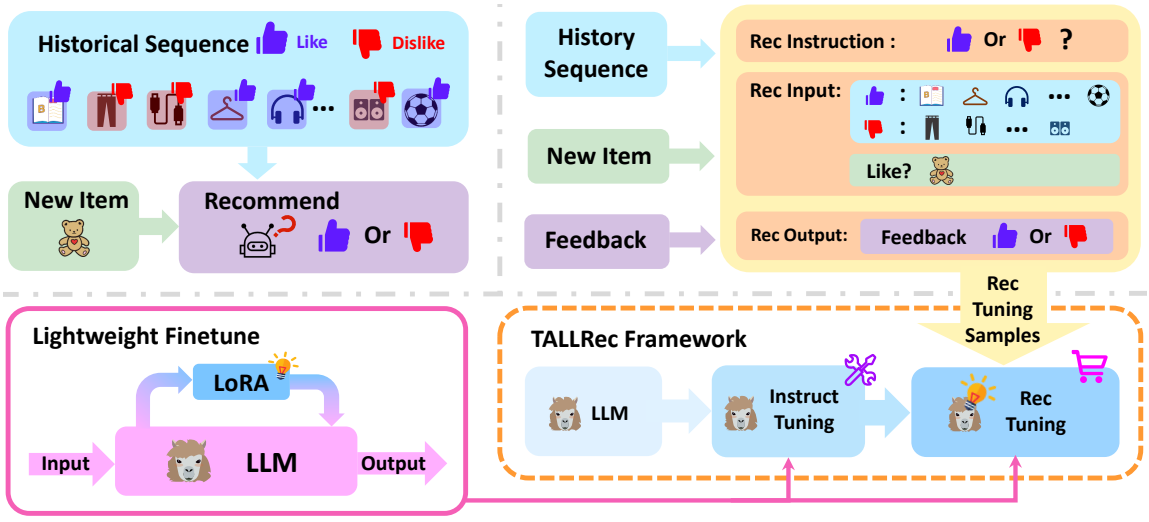

Fig. 2. Illustration of the TALLRec framework. In the upper left corner, we present an instance of sequential recommendation, where we leverage the user's interaction history to forecast their interest in a forthcoming item. In the upper right corner, we illustrate the approach for organizing the recommended data in such a scenario into instruction data for rec-tuning. The lower section of the figure showcases our proposed TALLRec framework. Notably, we employ lightweight fine-tuning technology to enhance the efficiency of our TALLRec framework.

interaction data to predict the user's preferences on novel items. The historical items interacted with by a user are represented as a sequence denoted by $[item_1, item_2, ..., item_n]$. Each item in the sequence contains the ID and textual information (*e.g.*, movie titles or book titles). Additionally, the sequence is accompanied by a user feedback sequence, denoted as $[rate_1, rate_2, ..., rate_n]$, where $rate_n \in \{1, 0\}$ indicates whether the user like the $item_n$ or not, similarly for others. In our setting, we are tasked with utilizing LLM denoted as $\mathcal{M}$ to construct an LRLM denoted as $\mathcal{M}_{rec}$, which can predict whether a new item (denoted as $item_{n+1}$) will be enjoyed by a user based on the recommendation task instructions and the user's historical interactions. In notation, the historical sequences combined with the new item are denoted as "*Rec Input*", the prediction of LRLM is represented as "*Rec Output*" and the "*Task Instruction*" for the recommendation task is represented as "*Rec Instruction*".

2.2 TALLRec Framework

In this subsection, we introduce the TALLRec framework, which aims to facilitate the effective and efficient alignment of the LLM with recommendations, particularly in low GPU memory consumption settings. Specifically, we first present the tuning methods, followed by the backbone selection.

TALLRec Tuning Stages. TALLRec comprises two tuning stages: instruction tuning and recommendation tuning (rec-tuning). The former stage is the common training process of LLM which enhances LLM's generalization ability, while the latter stage emulates the pattern of instruction tuning and fine-tunes the model for the recommendation task. For the former, we employ the self-instruct data made available by Alpaca [44] to train our model. Specifically, we utilize the conditional language modeling objective during the instruction tuning, as exemplified in the Alpaca

repository[1]. Formally,

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

where x and y represent the "*Instruction Input*" and "*Instruction Output*" in the self-instruct data, respectively, y_t is the t-th token of the y, $y_{<t}$ represents the tokens before y_t, Φ is the original parameters of $\mathcal{M}$, and $\mathcal{Z}$ is the training set.

For recommendation tuning, we format recommendation data into a pattern of instruction tuning. As illustrated in Figure 2, we begin by composing an "*Rec Instruction*" that directs the model to determine whether the user has a favorable disposition towards the target item based on their expressed preferences and dislikes, and to respond with a binary answer of "Yes" or "No". Secondly, we transform the "*Rec Input*" data into a natural language format. In detail, we begin by categorizing the historical interaction items into two groups based on their ratings: items that the user liked and items that the user did not like. We then combine these two groups with the target item to be recommended. We finally express this information in natural language to format a final "*Rec Input*" – "User Preference: $item_1$, $item_4$, ..., $item_n$. User Unpreference: $item_2$, $item_3$, ..., $item_{n-1}$. Whether the user will enjoy the target movie/book: $item_{n+1}$" – as shown in Figure 2. Then, we convert the feedback of the user to the new item to "Yes./No." as an "*Rec Output*". Ultimately, we merge the aforementioned "*Rec Instruction*" and "*Rec Input*" to create a "*Instruction Input*" for rec-tuning. We then utilize the resulting "*Rec Output*" as the "*Instruction Output*" for rec-tuning, and train the LLM on these rec-tuning samples in a similar way to the instruction tuning stage, to build an LRLM.

Lightweight Finetuning. When training the LLM, directly fine-tuning the model is computationally intensive and time-consuming. Given these factors, we propose to adopt a lightweight fine-tuning strategy to execute both instruction tuning and rec-tuning. The central premise of lightweight fine-tuning is that contemporary language models may possess an excessive number of parameters, and their information is concentrated on a low intrinsic dimension [19]. Consequently, we can achieve comparable performance to that of the entire model by fine-tuning only a small subset of parameters [18, 26, 28]. Specifically, we employ LoRA [19], which involves freezing the pre-trained model parameters and introducing trainable rank decomposition matrices into each layer of the Transformer architecture to facilitate lightweight fine-tuning. Therefore, by optimizing rank decomposition matrices, we can efficiently incorporate supplementary information while maintaining the original parameters in a frozen state. In total, the final learning objective can be computed as:

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

where Θ is the LoRA parameters and we only update LoRA parameters during the training process. Through LoRA, we can complete training with only one-thousandth of the original LLM parameters to complete the training process [19].

Backbone Selction. At present, there are large amounts of LLMs released, such as GPT series, PaLM, CHinchilla, and LLaMA [2, 3, 16, 45]. Among these, a considerable number of LLMs (such as PaLM and Chinchilla) do not provide access to their model parameters or APIs, rendering them challenging to utilize for research or other applications. Additionally, data security concerns are significant issues in the recommendation field. Consequently, the utilization of third-party APIs (such as ChatGPT and text-davinci-003) to leverage LLMs necessitates further discussion. To replicate the issues that require consideration in real-world recommendation scenarios, we intend to simulate the practical utilization of a public LLM and update its parameters for recommendation purposes. After careful consideration, we have opted

[1]https://github.com/tloen/alpaca-lora

to conduct experiments using LLMs-LLaMA, which is presently the best-performing open-source LLM, and whose training data is also publicly available [45].

3 EXPERIMENTS

In this section, we conduct experiments to answer the following research questions:

- **RQ1:** How is the performance of the proposed method compared with current LLM-based and traditional recommendation methods in scenarios with limited training interactions?
- **RQ2:** How do the components of the proposed TALLRec affect its effectiveness?
- **RQ3:** How well does the proposed method enable the recommender to generalize to different (even unseen) domains?

3.1 Experimental Settings

3.1.1 Dataset. We conduct experiments on two separate datasets from movie and book domains to assess recommendation methods.

- **Movie.** This is a refined dataset derived from the MovieLens100K benchmark dataset, which comprises user ratings on movies that range from one to five and comprehensive textual descriptions of movies such as "title" and "director." Specifically, we process the original dataset by sampling the most recent 10,000 interactions as prediction targets and allocating them into training, validation, and testing sets with a ratio of 8:1:1. To construct a full data sample for each prediction target item of a user, ten preceding user interactions are retained as historical interactions. Furthermore, we convert the rating into a binary label using a threshold of 3, where ratings exceeding 3 are labeled as "like" ('1'), while those below it are labeled as "dislike" ('0').
- **Book.** This pertains to a dataset of book recommendations obtained from the BookCrossing dataset [67]. The BookCrossing dataset comprises user ratings of books on a scale from 1-10, and textual descriptions of books, such as the information of 'Book-Author' and 'Book-Title'. To create the utilized dataset, we randomly select an item interacted by a user as the prediction target item for the user, and sample 10 remaining items the user interacted with as historical interactions for the target item[2]. This sampling process is repeated for every user in the original dataset. Subsequently, we partition the resulting data into training, validation, and testing sets with the same ratio of 8:1:1. Additionally, we binarize the ratings by applying a threshold of 5.

3.1.2 Few-shot Training Setting. To evaluate the recommendation performance of different methods with severely limited training data, we adopt a few-shot training paradigm where only a limited number of samples randomly selected from the training set are used for model training. This setting is referred to as 'K-shot' training setting, where K represents the number of training samples used. By setting an extremely small value for K, such as 64, we could test whether a method can acquire recommendation capability rapidly with severely limited training data.

3.1.3 Baseline. To provide a comprehensive evaluation of our TALLRec, we compare it against both LLM-based and traditional recommendation methods:

•LLM-based methods. We compare our method to current LLM-based recommendation methods that use In-context Learning to directly generate recommendations, as demonstrated in works such as [11] and [46]. To ensure a fair comparison, we align these methods with our task by using the same instructions as TALLRec. Specifically, we perform In-context Learning on different LLMs: 1) *Alpaca-LoRA*, 2) *Text-Davinvi-002*, 3) *Text-Daviniv-003*, and 4) *ChatGPT*, where

[2]This dataset lacks the timestamp information, and we thus construct historical interaction by random sampling.

Alpaca-LoRA is a model for reproducing Alpaca results based on the LLaMA model using LoRA and instruction tuning, and the later three are GPT series models provided by OpenAI.

•**Traditional methods.** Since our approach utilizes historical interactions to predict the subsequent interaction, which is similar to the sequential recommendation, we intend to compare it with the following sequential recommendation methods:

- **GRU4Rec [15].** This is an RNN-based sequential recommender, which utilizes GRU to encode the historical sequence.
- **Caser [43].** This is a CNN-based sequential recommender, which embeds a sequence of recent items as an 'image' and learns sequential patterns using horizontal and vertical convolution filters. The number of horizontal filters is fixed at 16, while its height is searched within {2,3,4}, and the number of vertical filters is set to 1.
- **SASRec [24].** This is an attention-based sequential recommender, which employs a multi-head self-attention model architecture similar to the decoder of Transformer to capture user preferences. As per the paper of SASRec, we set the number of heads to one.
- **DROS [60].** This is a current state-of-the-art sequential recommendation method, which deals with various recommendation out-of-distribution issues to enhance the generation performance by leveraging the distributionally robust optimization technique. We use the version of DROS implemented on GRU4Rec, provided by the authors[3].

The default implementation of these methods relies solely on item ID information to construct the recommender model. However, in our setting, we assume item text descriptions are available, and our method would utilize these text descriptions to generate recommendations. To ensure fair comparisons between our approach and the baseline methods, we further consider comparing the following variants of GRU4Rec and DROS:

- **GRU-BERT.** This is a variant of GRU4Rec that incorporates a pre-trained LLM model BERT [6] to leverage the text descriptions of items. Specifically, we feed the item text descriptions into BERT and concatenate the resulting CLS embedding, which encodes text descriptions, with the initial item embedding of GRU4Rec. We take the combined embedding as the item representation used by GRU4Rec to generate recommendation predictions.
- **DROS-BERT.** This variant of DROS is similarly modified to incorporate BERT, allowing it to utilize the text information of items.

3.1.4 Evaluation Metric. Given the similarity between our setting and explicit recommendation, which involves predicting user interest in a given target item, we adopt the conventional evaluation metric of explicit recommendation: Area Under the Receiver Operating Characteristic (AUC), as our evaluation metric. Specifically, we compute the AUC score using the Scikit-learn package [35].

3.1.5 Implementation Details. We employ Python 3.10 and PyTorch 2.0 [34] to implement all methods. To ensure uniform sequence lengths, we apply padding to sequences with a historical interaction length below a predetermined threshold (10), using the last interacted item as the padding item. Our method leverages both historically interacted items and corresponding user feedback as model input to capture user preference. To enable equitable comparison, for baselines, we assign each feedback an embedding and concatenate it with the item embedding, enabling the direct utilization of the user feedback information at model input for user interest modeling. For all methods, we optimize model parameters using the Adam optimizer at a default learning rate of 1e-3 with MSE loss as the optimization objective. We use weight decay for all methods, searching values in the range of [1e-3, 1e-4, 1e-5, 1e-6, 1e-7]. Regarding baselines' specific hyperparameters, we adhered to their respective papers' settings. For GRU-BERT and DROS-BERT,

[3]https://github.com/YangZhengyi98/DROS

Table 2. Performance comparison between conventional sequential recommendation baselines and TALLRec under different few-shot training settings. The reported result is the AUC multiplied by 100, with boldface indicating the highest score.

		Baseline						Ours
	Few-shot	GRU	Caser	SASRec	DROS	GRU-BERT	DROS-BERT	TALLRec
movie	16	49.07	49.68	50.43	50.76	50.85	50.21	**67.24**
	64	49.87	51.06	50.48	51.54	51.65	51.71	**67.48**
	256	52.89	54.20	52.25	54.07	53.44	53.94	**71.98**
book	16	48.95	49.84	49.48	49.28	50.07	50.07	**56.36**
	64	49.64	49.72	50.06	49.13	49.64	48.98	**60.39**
	256	49.86	49.57	50.20	49.13	49.79	50.20	**64.38**

we utilize the version of BERT-base released by Hugging Face[4] to extract text information, set the number of GRU layers to 4, and employ a large hidden size of 1024, aiming at aligning with BERT's embedding size. Furthermore, given that the predicted label is binary, we apply the Sigmoid activation function to the output of the baselines to determine the probability of user preference. Lastly, we run all methods five times with different random seeds and report the averaged results.

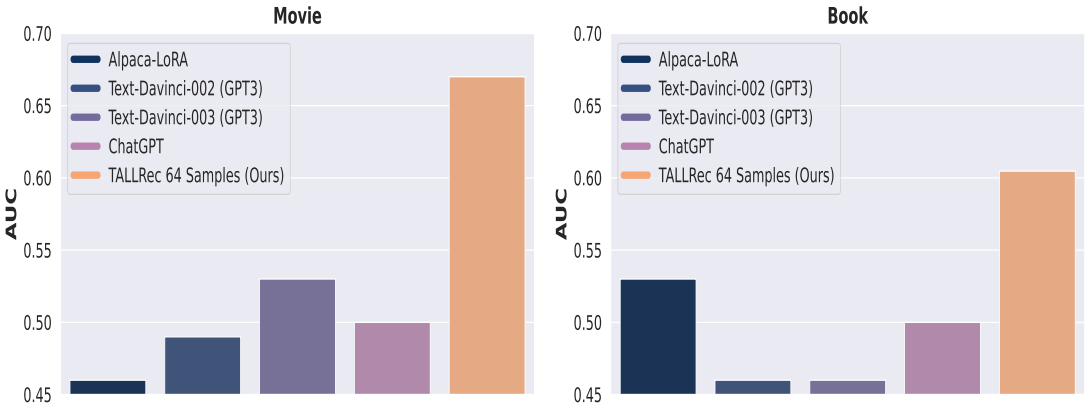

Fig. 3. Performance comparison between LLM-based baselines and TALLRec, where TALLRec is trained on only 64 samples (*i.e.*, in the 64-shot training setting).

3.2 Performance Comparison (RQ1)

We aim to investigate the recommendation performance of various methods under the few-shot training setting, which enables us to evaluate their ability to quickly inspire an effective recommendation capability with limited training samples. To comprehensively evaluate the performance of our method, we compare our proposed method with both traditional and recent LLM-based recommendation approaches. Our evaluation results against traditional methods are

[4]https://huggingface.co/bert-base-uncased

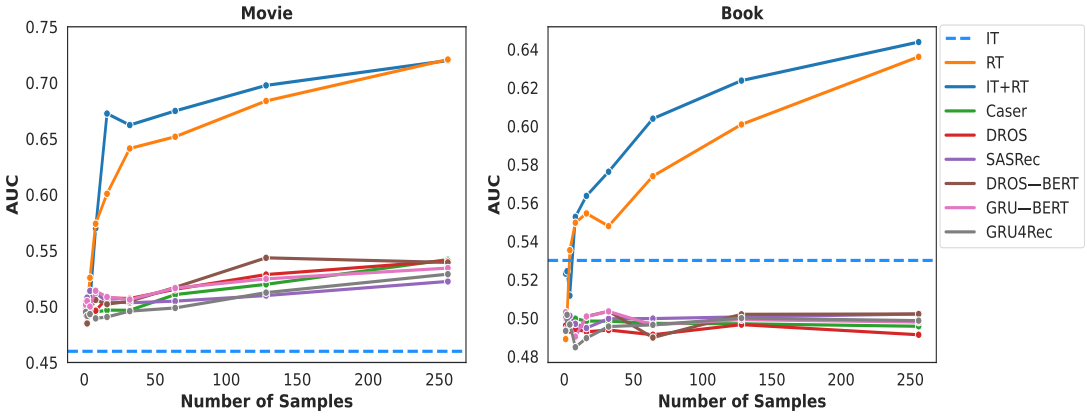

Fig. 4. Performance tendency of TALLRec's variants and conventional sequential recommendation methods *w.r.t.* the number of training samples used, ranging from 1 to 256. TALLRec has three variants: "IT" for instruction tuning only, "RT" for recommendation tuning only, and "IT + RT" for the full version.

presented in Table 2, while the comparison against LLM-based methods is depicted in Figure 3. Based on the figure and table, we draw the following observations:

- Our method significantly outperforms both traditional and recent LLM-based methods in the few-shot setting. This verifies the superiority of aligning the LLM to serve as a recommender through our TALLRec framework, which successfully leverages the know-rich and compositional generalization capabilities of the LLM for the recommendation.
- LLM-based methods perform similarly to random guessing with AUC values close to 0.5. However, the LRLM trained with our TALLRec achieves significant improvements over them. These results verify there is a considerable gap between recommendation and language tasks, and show the importance of recommendation data and recommendation tuning in inspiring the recommendation capability of LLMs.
- Traditional methods consistently yield AUC scores around 0.5 under our few-shot training settings, indicating that they also perform similarly to randomly guessing. This implies that traditional methods are incapable of quickly learning the recommendation capability with limited training samples.
- GRU-BERT and DROS-BERT, which are traditional recommendation methods enhanced with pre-trained language models (BERT), show no improvement over their initial methods and perform no better than random guessing. However, our TALLRec method demonstrates significantly superior and valid performance. These findings suggest that aligning the pre-trained language model directly with the recommendation task is better suited to unleash its potential for the recommendation, avoiding being restricted by the traditional recommenders.

3.3 Ablation Study (RQ2)

To demonstrate the efficacy of each component in the proposed TALLRec framework, encompassing the initial instruction tuning and recommendation tuning, we conduct a comparative analysis. Specifically, we compare the performance of three variants of TALLRec, including: "IT", "RT" and "IT+RT" versions, where "IT" denotes the version only conducting the initial instruction tuning, "RT" reflects solely implementing the recommendation tuning, and"IT+RT" signifies

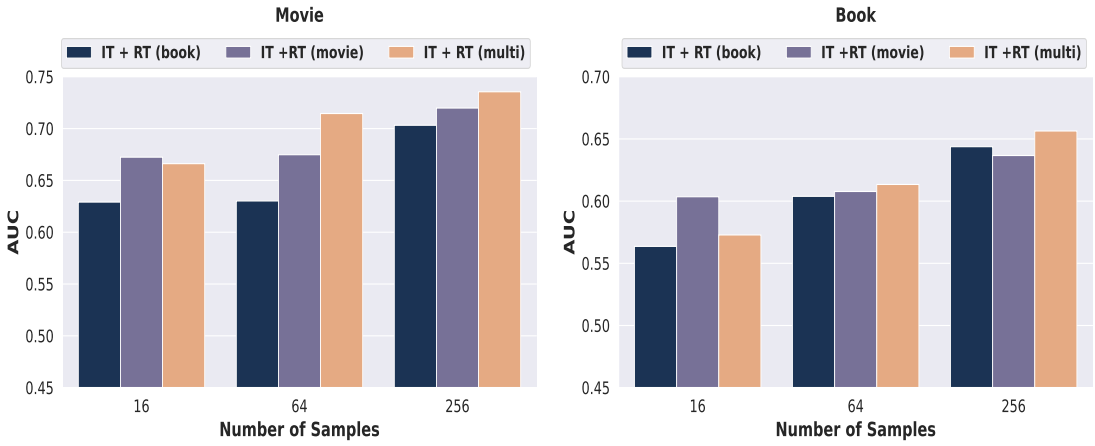

Fig. 5. Cross-domain performance of LRLMs trained via TALLRec on book data ('IT+RT (book)'), movie data ('IT+RT (movie)'), and both book and movie data ('IT+RT (multi)'). The left figure shows the testing results on movie data, while the right figure shows the testing results on book data.

the full version. Additionally, we vary the K of the few-shot setting, *i.e.,* the number of training samples utilized, to investigate the impact of the number of training samples on these variants and baselines. We summarize the results in Figure 4, where we have the following observations:

- Comparing "RT" and "IT+RT" with "IT", we observe that the variants with the recommendation tuning bring distinct performance improvements. This finding verifies the effectiveness and necessity of recommendation tuning in inspiring the LLM's recommendation capability.
- When the number of training samples is extremely limited ($\leq$ 128), "IT+RT" generally outperforms "RT". This observation confirms that instruction tuning can enhance the LLM's generalization ability to swiftly adapt to new tasks, and combining it with recommendation tuning could further enhance the efficacy of our recommendation tuning in scenarios with minimal training data.
- Our method exhibits the ability to inspire recommendation capabilities quickly with less than 50 training samples, outperforming traditional recommendation methods. This finding verifies the superiority of aligning the knowledge-rich and compositional generalization capabilities of LLMs with recommendation in few-shot training settings, showcasing the great potential of our proposal in data-limited recommendation scenarios.

3.4 Cross-domain Generalization Analyses (RQ3)

In our previous experiments, we evaluate the proposed method solely within the same domain as the training data. To investigate whether the Large Recommendation Language Model (LRLM) obtained by our TALLRec framework can exhibit universal recommendation generalization ability, specifically the ability to generalize well across different domains, we conducted further experiments with the cross-domain evaluation. Specifically, we trained three LRLMs using the TALLRec framework with different datasets, including 1) " IT+RT (book)", trained solely on the Book dataset; 2) " IT+RT (movie)", trained solely on the Movie dataset; and 3) " IT+RT (multi)", trained on both the Book and Movie datasets. We train these models sill under multiple few-shot training settings (K=16, 64, 258), and evaluate each of them

on the testing sets of both Book and Movie domains. The evaluation results are summarized in Figure 5, where we draw the following conclusions:

- Our TALLRec framework demonstrates remarkable cross-domain generalization ability. For instance, after training only on movie data, the obtained model exhibits strong performance on book data, comparable to the model trained exclusively on book data. This is quite impressive and suggests that the model learns the task itself rather than merely fitting the data like traditional recommenders.
- Our proposed TALLRec can utilize data from two different domains simultaneously to improve the recommendation performance across domains. For example, we can see that on the movie dataset, using a mixture of book and movie data performs better than training solely on movie data in most cases. This indicates that TALLRec can seamlessly incorporate data from different domains to enhance its performance.

4 RELATED WORK

In this section, we mainly talk about the development of related areas to our work.

4.1 Language Models for Recommendation

Currently, there have been several attempts to integrate language models (LMs) with recommendation systems. However, some of these attempts still rely on traditional user/item IDs to represent users/items, despite incorporating LMs [12, 27]. This approach fails to leverage the semantic understanding capabilities that LMs inherently possess, which could potentially enhance the accuracy and effectiveness of recommendation systems. Furthermore, there have been attempts to utilize LMs to encode the natural language information of users/items, such as reviews, and incorporate them as part of the embedding of users/items [17]. In addition, other methods either utilize an undisclosed model that already possesses preliminary recommendation capabilities [5], or employ small models to train on large-scale downstream task data [65]. Moreover, the aforementioned models are also limited to small models, while this paper is on an orthogonal direction about how to adapt large language models to recommendation tasks.

In the field of recommendation systems, there is currently little research on the application of LLMs in recommendation scenarios. The only work we know is to utilize the multi-round interaction ability of GPT3.5 series models and apply In-context Learning [11, 46]. In detail, Chat-Rec [11] endeavors to harness the robust multi-round interaction capabilities of ChatGPT and link the ChatGPT with traditional recommendation models (e.g. MF [25], LightGCN [14]) to formulate a conversational recommendation system. NIR shares a similar concept with Chat-Rec, albeit with a different approach. NIR [46] employs conventional recommendation models to generate a pool of potential items, which are subsequently subjected to a three-stage multi-step prompting process for re-ranking.

4.2 Sequential Recommendation

Our setup is close to sequential recommendation which aims to infer the user's next preference based on the historical sequence of user behaviours [10, 49]. In the early time, the Markov chain plays an important role in sequential recommendation [13, 31, 39, 48]. Recently, the deep learning-based method has become mainstream to model user behaviours. Huge amount of work using different kinds of neural network structures, like RNN-based [4, 8, 15], CNN-based [43, 59, 62], and attention-based [24, 58, 63], to model the user behavior. But limited by the traditional paradigm of using IDs to represent users or items, they cannot fast adapt and generalize. Thus, some works focus on the generalization ability of sequential recommendation models by pretraining [32, 61], data augmentation [37, 38, 51, 57],

debiasing [7, 52, 64, 66], and robust optimization [55, 60]. However, they still suffer from the inherent flaw in traditional ID-based representations, and are hard to adapt quickly to a new recommendation scenario.

5 CONCLUSION

In this paper, we first attempt to enhance the recommendation ability of LLM through the fine-tuning approach and investigate the feasibility of using LLM for the recommendation. Our initial findings reveal that even the existing best LLM models do not perform well in recommendation tasks. To address this issue, we propose a TALLRec framework that can efficiently align LLM with recommendation tasks through two stages: instruction tuning and rec tuning. Our experimental results demonstrate that the LRLM trained using our TALLRec framework outperforms traditional models and exhibits strong cross-domain transfer capabilities. Moving forward, we first plan to explore more efficient methods to activate the recommendation ability of large models and develop an LLM that can handle multiple recommendation tasks simultaneously. We will also follow previos work [50] and dedicate to exploring novel directions of generative recommendations.

REFERENCES

[1] Greg Brockman, Mira Murati, Peter Welinder, and OpenAI. 2022. OpenAI API. https://openai.com/blog/openai-api

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[3] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

[4] Qiang Cui, Shu Wu, Qiang Liu, Wen Zhong, and Liang Wang. 2020. MV-RNN: A Multi-View Recurrent Neural Network for Sequential Recommendation. *IEEE Trans. Knowl. Data Eng.* 32, 2 (2020), 317–331. https://doi.org/10.1109/TKDE.2018.2881260

[5] 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).

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[7] Sihao Ding, Fuli Feng, Xiangnan He, Jinqiu Jin, Wenjie Wang, Yong Liao, and Yongdong Zhang. 2022. Interpolative Distillation for Unifying Biased and Debiased Recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 40–49. https://doi.org/10.1145/3477495.3532002

[8] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*. 152–160.

[9] 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. *CoRR* abs/2303.03378 (2023). https://doi.org/10.48550/arXiv.2303.03378 arXiv:2303.03378

[10] 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. https://doi.org/10.1145/3426723

[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] 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.

[14] 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.

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

[16] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556* (2022).

[17] 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 *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 585–593.

[18] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 2790–2799. http://proceedings.mlr.press/v97/houlsby19a.html

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=nZeVKeeFYf9

[20] Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization. *arXiv preprint arXiv:2212.12017* (2022).

[21] Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the Benefits of Training Expert Language Models over Instruction Tuning. *CoRR* abs/2302.03202 (2023). https://doi.org/10.48550/arXiv.2302.03202 arXiv:2302.03202

[22] Vitor Jeronymo, Luiz Henrique Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto de Alencar Lotufo, Jakub Zavrel, and Rodrigo Frassetto Nogueira. 2023. InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. *CoRR* abs/2301.01820 (2023). https://doi.org/10.48550/arXiv.2301.01820 arXiv:2301.01820

[23] Matthew Jin, Syed Shahriar, Michele Tufano, Xin Shi, Shuai Lu, Neel Sundaresan, and Alexey Svyatkovskiy. 2023. InferFix: End-to-End Program Repair with LLMs. *CoRR* abs/2303.07263 (2023). https://doi.org/10.48550/arXiv.2303.07263 arXiv:2303.07263

[24] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. 197–206.

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

[26] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 3045–3059. https://doi.org/10.18653/v1/2021.emnlp-main.243

[27] Lei Li, Yongfeng Zhang, and Li Chen. 2023. Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems* 41, 4 (2023), 1–26.

[28] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 4582–4597. https://doi.org/10.18653/v1/2021.acl-long.353

[29] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110* (2022).

[30] Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2021. Few-shot learning with multilingual language models. *arXiv preprint arXiv:2112.10668* (2021).

[31] Tariq Mahmood and Francesco Ricci. 2007. Learning and adaptivity in interactive recommender systems. In *Proceedings of the ninth international conference on Electronic commerce*. 75–84.

[32] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. 2018. Perceive Your Users in Depth: Learning Universal User Representations from Multiple E-commerce Tasks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 596–605. https://doi.org/10.1145/3219819.3219828

[33] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[35] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python.

[36] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction Tuning with GPT-4. *CoRR* abs/2304.03277 (2023). https://doi.org/10.48550/arXiv.2304.03277 arXiv:2304.03277

[37] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2021. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. *CoRR* abs/2110.05730 (2021). arXiv:2110.05730 https://arxiv.org/abs/2110.05730

[38] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang (Eds.). ACM, 813–823. https://doi.org/10.1145/3488560.3498433

[39] 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.

[40] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207* (2021).

[41] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761* (2023).

[42] Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. 2022. LM-Nav: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand (Proceedings of Machine Learning Research, Vol. 205)*, Karen Liu, Dana Kulic, and Jeffrey Ichnowski (Eds.). PMLR, 492–504. https://proceedings.mlr.press/v205/shah23b.html

[43] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.

[44] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LlaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[46] Lei Wang and Ee-Peng Lim. 2023. Zero-Shot Next-Item Recommendation using Large Pretrained Language Models. *arXiv preprint arXiv:2304.03153* (2023).

[47] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. *CoRR* abs/2303.07678 (2023). https://doi.org/10.48550/arXiv.2303.07678 arXiv:2303.07678

[48] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*. 403–412.

[49] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet A. Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 6332–6338. https://doi.org/10.24963/ijcai.2019/883

[50] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2023. Generative recommendation: Towards next-generation recommender paradigm. *arXiv preprint arXiv:2304.03516* (2023).

[51] Ziyang Wang, Huoyu Liu, Wei Wei, Yue Hu, Xian-Ling Mao, Shaojian He, Rui Fang, and Dangyang Chen. 2022. Multi-level Contrastive Learning Framework for Sequential Recommendation. *CoRR* abs/2208.13007 (2022). https://doi.org/10.48550/arXiv.2208.13007 arXiv:2208.13007

[52] Zhenlei Wang, Shiqi Shen, Zhipeng Wang, Bo Chen, Xu Chen, and Ji-Rong Wen. 2022. Unbiased Sequential Recommendation with Latent Confounders. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (Eds.). ACM, 2195–2204. https://doi.org/10.1145/3485447.3512092

[53] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).

[54] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).

[55] Hongyi Wen, Xinyang Yi, Tiansheng Yao, Jiaxi Tang, Lichan Hong, and Ed H. Chi. 2022. Distributionally-robust Recommendations for Improving Worst-case User Experience. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (Eds.). ACM, 3606–3610. https://doi.org/10.1145/3485447.3512255

[56] Ted Xiao, Harris Chan, Pierre Sermanet, Ayzaan Wahid, Anthony Brohan, Karol Hausman, Sergey Levine, and Jonathan Tompson. 2022. Robotic Skill Acquisition via Instruction Augmentation with Vision-Language Models. *CoRR* abs/2211.11736 (2022). https://doi.org/10.48550/arXiv.2211.11736 arXiv:2211.11736

[57] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 1259–1273. https://doi.org/10.1109/ICDE53745.2022.00099

[58] Chengfeng Xu, Jian Feng, Pengpeng Zhao, Fuzhen Zhuang, Deqing Wang, Yanchi Liu, and Victor S. Sheng. 2021. Long- and short-term self-attention network for sequential recommendation. *Neurocomputing* 423 (2021), 580–589. https://doi.org/10.1016/j.neucom.2020.10.066

[59] An Yan, Shuo Cheng, Wang-Cheng Kang, Mengting Wan, and Julian J. McAuley. 2019. CosRec: 2D Convolutional Neural Networks for Sequential Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 2173–2176. https://doi.org/10.1145/3357384.3358113

[60] 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. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2023).

[61] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1469–1478. https://doi.org/10.1145/3397271.3401156

[62] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman (Eds.). ACM, 582–590. https://doi.org/10.1145/3289600.3290975

[63] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 4320–4326. https://doi.org/10.24963/ijcai.2019/600

[64] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 11–20. https://doi.org/10.1145/3404835.3462875

[65] Zizhuo Zhang and Bang Wang. 2023. Prompt Learning for News Recommendation. *arXiv preprint arXiv:2304.05263* (2023).

[66] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 2980–2991. https://doi.org/10.1145/3442381.3449788

[67] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving Recommendation Lists through Topic Diversification. In *Proceedings of the 14th International Conference on World Wide Web* (Chiba, Japan) *(WWW '05)*. Association for Computing Machinery, New York, NY, USA, 22–32. https://doi.org/10.1145/1060745.1060754