

Distillation Matters: Empowering Sequential Recommenders to Match the Performance of Large Language Models

Yu Cui
Zhejiang University
Hangzhou, China
12321228@zju.edu.cn

Feng Liu
OPPO Co Ltd
Shenzhen, China
liufeng4@oppo.com

Pengbo Wang
University of Electronic Science and
Technology of China
Chengdu, China
2021080902021@std.uestc.edu.cn

Bohao Wang
Zhejiang University
Hangzhou, China
bohao.wang@zju.edu.cn

Heng Tang
Zhejiang University
Hangzhou, China
tangheng23@zju.edu.cn

Yi Wan
OPPO Co Ltd
Shenzhen, China
wanyan@oppo.com

Jun Wang
OPPO Co Ltd
Shenzhen, China
wangjun7@oppo.com

Jiawei Chen
Zhejiang University
Hangzhou, China
sleepyhunt@zju.edu.cn

Abstract

Owing to their powerful semantic reasoning capabilities, Large Language Models (LLMs) have been effectively utilized as recommenders, achieving impressive performance. However, the high inference latency of LLMs significantly restricts their practical deployment. To address this issue, this work investigates knowledge distillation from cumbersome LLM-based recommendation models to lightweight conventional sequential models. It encounters three challenges: 1) the teacher’s knowledge may not always be reliable; 2) the capacity gap between the teacher and student makes it difficult for the student to assimilate the teacher’s knowledge; 3) divergence in semantic space poses a challenge to distill the knowledge from embeddings.

To tackle these challenges, this work proposes a novel distillation strategy, DLLM2Rec, specifically tailored for knowledge distillation from LLM-based recommendation models to conventional sequential models. DLLM2Rec comprises: 1) *Importance-aware ranking distillation*, which filters reliable and student-friendly knowledge by weighting instances according to teacher confidence and student-teacher consistency; 2) *Collaborative embedding distillation* integrates knowledge from teacher embeddings with collaborative signals mined from the data. Extensive experiments demonstrate the effectiveness of the proposed DLLM2Rec, boosting three typical sequential models with an average improvement of 47.97%, even enabling them to surpass LLM-based recommenders in some cases.

CCS Concepts

• Information systems → Recommender systems.

Keywords

Sequential Recommendation, Large language Model, Knowledge Distillation

ACM Reference Format:

Yu Cui, Feng Liu, Pengbo Wang, Bohao Wang, Heng Tang, Yi Wan, Jun Wang, and Jiawei Chen. 2018. Distillation Matters: Empowering Sequential Recommenders to Match the Performance of Large Language Models. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Large Language Models (LLMs) have showcased remarkable capabilities in content comprehension, generation, and semantic reasoning [1, 5, 55]. Recently, LLMs have sparked a surge of interest within the domain of Recommender Systems (RS). Various research efforts have been devoted to harnessing LLMs to augment traditional recommendation models, serving as encoders for user/item features or as supplementary knowledge bases [20, 21, 45, 46]. To fully exploit the reasoning ability of LLMs in RS, another line of research is to directly prompt or fine-tune LLMs to function as specialized recommenders. Owing to their inherent semantic reasoning capabilities, these LLM-based recommendation methods have achieved impressive performance. For instance, as shown in Table 1, the representative model BIGRec [3] has demonstrated approximately 41.44% improvements on average over the state-of-the-art conventional sequential model (*i.e.*, DROS [69]) on the typical Amazon Games and Toys datasets.

Despite their effectiveness, LLM-based recommenders face serious inference inefficiency issues, posing substantial challenges to their practical applications. For example, as Table 1 shows, the widely used LLaMA2-7B model requires an astonishing three hours

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/XXXXXXX.XXXXXXX>

to perform a single inference for tens of thousands of users with 4x A800 GPUs. This inefficiency is exacerbated when scaling up to serve millions of users concurrently in practical industrial RS, where responses are required within seconds. This motivates a crucial research question: *how can we maintain low inference latency as conventional recommenders while leveraging the excellent performance of LLM-based recommenders?*

To tackle this challenge, we propose employing knowledge distillation (KD) for acceleration — *i.e.*, distilling the knowledge from a complex LLM-based recommendation model (teacher) to a lightweight conventional recommendation model (student). KD has been successfully applied in many domains [2, 8, 19], and has been proven effective in transferring knowledge from a large model to a smaller one. This strategy could capitalize on the effectiveness of LLM-based recommenders while maintaining low inference costs. It also potentially integrates the capabilities of conventional models in capturing collaborative signals with the semantic reasoning prowess of LLMs. While the ideal is promising, distillation is non-trivial due to the fundamentally different mechanisms between the teacher and student models. LLMs primarily rely on content understanding and capturing semantic correlations for making recommendations, whereas conventional models depend on collaborative signals derived from users’ historical behaviors. This divergence introduces several challenges:

1) **Teacher Knowledge Reliability:** LLM-based models may not consistently outperform conventional models in all cases. Our empirical studies suggest that in over 30% of cases, a conventional model could even outperform an LLM-based model, indicating that the knowledge from the teacher is not always reliable. Moreover, the LLM may encounter the issue of notorious *hallucination* and generate poor recommendations.

2) **Model Capacity Gap:** As demonstrated by recent work on KD [10], the substantial difference in model size often makes it difficult for the student to assimilate the teacher’s knowledge. Given the simple architecture of the conventional models, they may struggle to fully inherit the semantic reasoning ability of the teacher, and overloading the student with teacher knowledge might impair its own ability in collaborative filtering.

3) **Semantic Space Divergence:** Aligning embeddings for distillation, which has been demonstrated effective, presents significant challenges for this problem. LLMs model users/items primarily based on content, while conventional models derive users/items embeddings from collaborative signals. The vast semantic differences between these approaches mean that directly aligning their embeddings can be counterproductive, potentially disrupting the student’s original embedding space and weakening its ability to capture collaborative signals.

To tackle these challenges, we propose DLLM2Rec, designed to effectively distill knowledge from LLM-based recommenders to conventional recommenders. DLLM2Rec contains:

1) **Importance-aware ranking distillation.** Rather than directly aligning the ranking lists between the teacher and student, we highlight reliable and student-friendly instances for distillation by introducing importance weights. This approach evaluates the semantic similarity between the responses given by LLMs and the target positive items, with less similarity indicating lower response quality and suggesting such instances should be downweighted in

Table 1: Recommendation performance and inference time-cost of BIGRec compared with DROS on Amazon Games and Toys datasets. Note that BIGRec is a typical LLM-based recommender with LLaMA-7B and DROS is a state-of-the-art sequential recommendation method.

Dataset	Model	HR@20	NDCG@20	Inference time
Games	DROS	0.0473	0.0267	1.8s
	BIGRec	0.0532	0.0341	2.3×10^4 s
	Gain	+12.47%	+27.72%	$-1.3 \times 10^6\%$
Toys	DROS	0.0231	0.0144	1.6s
	BIGRec	0.0420	0.0207	1.1×10^4 s
	Gain	+81.82%	+43.75%	$-6.8 \times 10^5\%$

distillation. Additionally, inspired by the “wisdom of the crowd”, we leverage the model consistency between student and teacher to evaluate the importance of an instance, prioritizing instances where diverse models agree on higher item rankings. Such instances are also relatively easy and friendly to the student models, helping the student to assimilate the knowledge from the teacher.

2) **Collaborative embedding distillation.** To bridge the semantic gap between the embedding spaces of the teacher and student, we employ a learnable projector (*e.g.*, MLPs) to map original embeddings from teachers to the student’s embedding space. Moreover, diverging from directly aligning the student embeddings with the teacher’s projected embeddings, we introduce a flexible offset term that captures collaborative signals, further integrated with the teacher’s projected embeddings to generate enriched student embeddings. This design effectively leverages the knowledge from the teacher while preserving its capacity to capture collaborative signals.

The main contributions of our work are summarized as follows:

- Highlighting the inference inefficiency issue of LLM-based recommendation model and advocating the use of knowledge distillation for acceleration.
- Proposing DLLM2Rec which leverages importance-aware ranking distillation and collaborative embedding distillation to transfer reliable and student-friendly knowledge from LLM-based models to conventional recommendation models.
- Conducting extensive experiments to demonstrate the effectiveness of DLLM2Rec, enabling lightweight conventional models to keep pace with sophisticated LLM-based models.

2 PRELIMINARIES

In this section, we elaborate on sequential recommendation, and introduce BIGRec, a representative LLM-based recommender.

2.1 Sequential Recommendation

This work focuses on sequential recommendation, which has secured a pivotal role in various modern recommendation systems and attracted significant research interest. In a sequential recommender system with a user set \mathcal{U} and an item set \mathcal{I} , a user’s historical interactions can be organized in chronological order $\mathbf{s}_u^t = (i_1, i_2, \dots, i_{t-1})$ where $i_k \in \mathcal{I}$ represents the k -th item that the user u interacted with. We remark that in this paper we may shorten the notation \mathbf{s}_u^t as \mathbf{s} for clear presentation. The task of

sequential recommendation is to predict the next item i_t that the user is likely to interact with.

Sequential Recommendation Model. Existing models in this domain primarily adopt representation learning paradigms. These methods first utilize an item encoder to map items' features x_i (e.g., IDs, titles) into their representations \mathbf{e}_i :

$$\mathbf{e}_i = \text{ItemEncoder}(x_i) \quad (1)$$

where $\text{ItemEncoder}(\cdot)$ can be implemented by various architectures, e.g., an embedding layer to encode item ID or BERT [11] to encode item text.

With the item embeddings, user behaviors can be further encoded by a sequential encoder:

$$\mathbf{e}_s = \text{SeqEncoder}(\mathcal{Z}_s) \quad (2)$$

where \mathcal{Z}_s denotes the encoded item embedding sequence of s , i.e., $\mathcal{Z}_s = (\mathbf{e}_{i_1}, \mathbf{e}_{i_2}, \dots, \mathbf{e}_{i_{t-1}})$. $\text{SeqEncoder}(\cdot)$ represents the sequence encoder and can be implemented by GRUs [18], Transformer [28], or other architectures.

Given the sequence and item embeddings, the final prediction \hat{y}_{si} can be generated via the dot product [42] or MLPs [43], which is then utilized to retrieve recommendations. Let i_s^* (shorten as i^*) be the ground-truth item of the sequence s that the user u will interact with at the next step, the model can be trained via various losses, e.g., binary cross-entropy [47]:

$$\mathcal{L}_r = - \sum_{s \in \Gamma} \left(\log \sigma(\hat{y}_{si^*}) + \sum_{j \in O^-} \log(1 - \sigma(\hat{y}_{sj})) \right) \quad (3)$$

where $\sigma(\cdot)$ denotes the Sigmoid function; Γ denotes the set of sequences used for model training; and O^- denotes the set of sampled negative items.

2.2 Brief on BIGRec

Recently LLM-based recommendation attracts great attention. Predominantly, this body of work formulates the recommendation task using natural language prompts and employs large language models to generate personalized recommendations [22]. This study simply take the representative model BIGRec [3] for empirical analysis. The selection of BIGRec is justified not only by its availability as an open-source tool but also by its demonstrated effectiveness. Furthermore, BIGRec embodies the fundamental elements of LLM-based recommendation and many methods can be considered as further extensions of such paradigm [50, 61]. It is also noteworthy that BIGRec has been employed by recent studies [39, 40] as a representative model for analysis.

To be specific, BIGRec organizes users' historical behaviors in natural language and employs instruction-tuning to fine-tune LLMs, as illustrated in Figure 1. During the inference stage, BIGRec generates item descriptions (e.g., titles) for recommendations. Considering that these descriptions may not always correspond to existing items, BIGRec incorporates a grounding paradigm that matches generated item descriptions to existing items based on content similarity. Formally, let \mathbf{z}_{g_s} and \mathbf{z}_i represent the token embeddings of generated descriptions and the descriptions of item i , respectively. BIGRec computes their L2 distance for grounding as follows:

$$d_{si} = \|\mathbf{z}_{g_s} - \mathbf{z}_i\|^2 \quad (4)$$

Table 2: The ratio of cases where BIGRec outperforms DROS to cases where BIGRec underperforms DROS.

Dataset	Condition	Relative Ratio
Games	BIGRec > DROS	53.90%
	BIGRec < DROS	46.10%
MovieLens	BIGRec > DROS	40.90%
	BIGRec < DROS	59.10%
Toys	BIGRec > DROS	66.67%
	BIGRec < DROS	33.33%

Table 3: Ratio of overlapped items in Top-20 recommendations between BIGRec and DROS. Additionally, we present the percentage of these items that are actual hits. For comparative analysis, we detail the values specific to items unique to either BIGRec's or DROS's recommendations.

Dataset	Rec. Space	Items Ratio	Hit Items
Games	BIGRec only	96.01%	0.21%
	DROS only	96.01%	0.18%
	Overlapped	3.99%	1.61%
MovieLens	BIGRec only	95.94%	0.19%
	DROS only	95.94%	0.35%
	Overlapped	4.06%	2.16%
Toys	BIGRec only	98.95%	0.17%
	DROS only	98.95%	0.08%
	Overlapped	1.05%	3.74%

Based on d_{si} , BIGRec ranks items and retrieve the K-nearest items as recommendations.

3 METHODOLOGY

In this section, we first outline the challenges associated with distilling knowledge from Large LLM-based recommendation models to conventional models (subsection 3.1). Following this, we delve into the specifics of our proposed DLLM2Rec (subsections 3.2).

3.1 Motivations

In this subsection, we aim to conduct a thorough analysis to elucidate the challenges of distillation, thereby motivating our proposed method. These challenges can be categorized into three aspects:

Teacher Knowledge Reliability. By examining the recommendation results from a typical LLM-based model, BIGRec, and a state-of-the-art sequential model, DROS, on three real-world datasets, we empirically discover that the LLM-based model may not consistently surpass the conventional model in all test cases. In fact, as depicted in Table 2, BIGRec may underperform DROS in over 30% of cases across all three datasets. This observation implies that the teacher's knowledge may not always be reliable and could potentially be detrimental. The reliability of the teacher's knowledge in distillation must be validated.

Model Capacity Gap. Recent research suggests that the performance of a student model diminishes as the gap in size between the teacher and student models increases [24]. This challenge is even more pronounced in our scenario, where the student model comprises a million-level parameters while the teacher model has billion-level parameters. Additionally, the teacher and student models employ fundamentally different recommendation mechanisms.

We notice a significant discrepancy in their recommended items — the average number of overlapped items in their Top-20 recommendations is less than 3.15% across the three datasets as shown in Table 3. It is implausible to expect the student to fully assimilate the teacher’s knowledge and fully inherit the teacher’s ability on semantic reasoning. Overloading the student with the teacher’s knowledge may even impair the student’s inherent capacity to capture collaborative signals. Our empirical study, as shown in Table 5, demonstrates that existing knowledge distillation strategies usually yield limited improvements and can sometimes even be counterproductive. Thus, the development of a distillation strategy that is friendly to the student model is crucial.

Semantic Space Divergence. It is noteworthy that LLM-based models characterize users/items mainly based on their contents, while conventional models derive users/items embeddings mainly from collaborative signals. It means the teacher and student adopt entirely different semantic frameworks. Blind alignment of their semantic spaces for distillation could prove counterproductive. As observed in Table 5, two representative knowledge distillation methods, Hint [2] and HTD [27], which distill through embeddings, often perform worse than the original student model without knowledge distillation. While embedding distillation has proven effective in many domains, it should be specifically designed for this task.

3.2 Proposed Distillation Strategy: DLLM2Rec

In order to address the aforementioned challenges, this work proposes DLLM2Rec, with leveraging importance-aware ranking distillation and collaborative embedding distillation.

3.2.1 Importance-aware Ranking Distillation. This module builds upon the conventional ranking distillation [53] while additionally introducing importance weights to emphasize reliable and student-friendly instances. Specifically, we employ the following distillation loss:

$$\mathcal{L}_d = - \sum_{s \in \Gamma} \sum_{i \in O^T} w_{si} \log \sigma(\hat{y}_{si}) \quad (5)$$

where O^T denotes the Top-K recommendations returned by the teacher model. In this work, we choose $K = 10$ as a default value, but this can be tuned for optimal performance. The objective is straightforward — we select the highly ranked items from the teacher as a positive to guide the learning of the student, so that these candidate items can also be recommended by the student. However, given that the teacher’s recommendations may not always be beneficial, we introduce an importance weight that considers the following three aspects:

1) **Position-aware weights** w_{si}^p . Inheriting from [53], ranking positions are also considered in DLLM2Rec. The motivation is from the ranking alignment that we would like to push a candidate item higher if the item also occupies a higher position in the teacher’s ranking list. Formally, we use:

$$w_{si}^p \propto \exp(-r_i/\beta) \quad (6)$$

where r_i denotes the position of item i in the ranking list returned by the teacher, and β is the hyperparameter adjusting the shape of the weight distribution.

2) **Confidence-aware weights** w_{si}^c . Given the importance of extracting reliable teacher knowledge, we leverage w_{si}^c to indicate

reliability. Specifically, we measure the quality of descriptions generated by LLMs by assessing the content distance between the generated descriptions and the content of the ground-truth item:

$$\begin{aligned} w_{si}^c &\propto \exp(-d_{si}^*/\beta) \\ d_{si}^* &= \|z_{g_s} - z_{i^*}\|^2 \end{aligned} \quad (7)$$

where d_{si}^* measures the embedding distance between the generated item description z_{g_s} and the target ground-truth item z_{i^*} , where the embeddings can be generated via LLMs encoder. A smaller distance suggests a higher quality of the generated description as it aligns more closely with the targets. Conversely, a larger gap suggests lower confidence, indicating that LLMs may risk generating incorrect or nonsensical information.

3) **Consistency-aware weights** w_{si}^s . Inspired by the “wisdom of the crowd”, we use model consistency between the student and teacher to assess the importance of an instance. As suggested by recent work on bagging [12, 70], when diverse models reach a consensus on one prediction, its reliability increases. In RS, our empirical studies (Table 3) also shows that the items that are concurrently recommended by teacher and student are more likely to be positive. This insight allows us to formulate consistency-aware weights as follows:

$$w_{si}^o = \begin{cases} 1, & i \in O^T \cap O^S \\ 0, & i \notin O^T \cap O^S \end{cases} \quad (8)$$

where O^T and O^S denote the sets of Top-K recommendation items returned by the teacher and student, respectively. We assign higher weights to those overlapping items (i.e., $i \in O^T \cap O^S$).

Another advantage for up-weighting those overlapped items is that they are relatively easy and friendly for student learning. By examining the gradient of the distillation loss:

$$\frac{\partial \mathcal{L}_d}{\partial \hat{y}_{si}} = w_{si} \sigma(-\hat{y}_{si}) \quad (9)$$

it is evident that instances with larger \hat{y}_{si} , i.e., higher positions in the student ranking lists, will have smaller gradient magnitudes. This suggests that higher-ranked instances are more easily assimilated by the student model, as the student does not require to make extensive change. Upweighting these instances makes the knowledge distillation process more conducive to student learning.

We integrate these three aspects into the ranking distillation with a simple linear combination:

$$w_{si} = \gamma_p \cdot w_{si}^p + \gamma_c \cdot w_{si}^c + \gamma_o \cdot w_{si}^o \quad (10)$$

where $\gamma_p, \gamma_c, \gamma_o$ are hyperparameters balancing their contributions. The ultimate objective of our DLLM2Rec is:

$$\mathcal{L} = \mathcal{L}_r + \lambda_d \mathcal{L}_d \quad (11)$$

where λ_d balances the contributions from the recommendation and distillation losses.

Interestingly, some recent work [31, 53] consider to up-weight the instances which has larger ranking discrepancy between student and teacher. This strategy is ineffective in this task, as it would increase the distillation unreliability and difficulty. Our DLLM2Rec adopts contrary strategy and would yield better performance as demonstrated in our experiments.

3.2.2 Collaborative Embedding Distillation. Embedding distillation has proven effective in many domains, yet it requires careful design

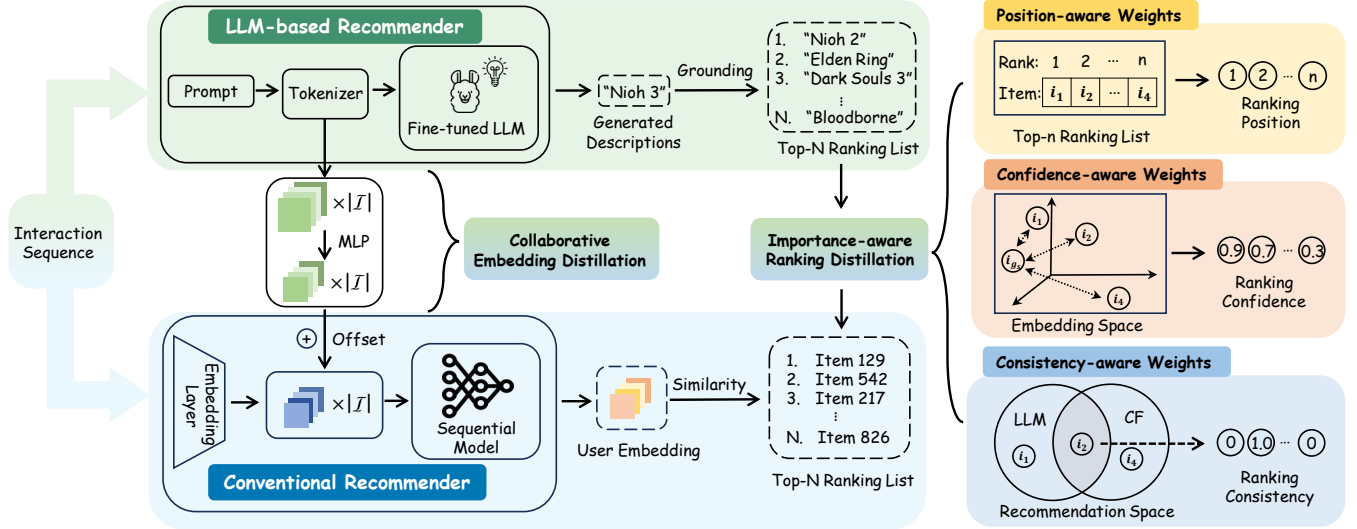


Figure 1: Illustration the proposed DLLM2Rec that distills the knowledge from the LLM-based recommenders to the conventional recommenders, with leveraging importance-aware ranking distillation and collaborative embedding distillation.

in this context, given that the teacher and student possess quite different semantic spaces. To tackle this, we adopt a collaborative paradigm. Specifically, we first employ a learnable projector (e.g., MLPs) to map original item embeddings from the teacher to the student’s embedding space to bridge the semantic gap:

$$z_i^p = g(z_i) \quad (12)$$

where z_i denotes the textual embedding of item i encoded by the LLM-based recommender; $g(\cdot)$ denotes a learnable projector function, which can be implemented via MLPs.

we further introduce a flexible offset term b_i for each item, which is integrated with the teacher’s projected embeddings to generate enriched student embeddings:

$$e_i^{new} = f(z_i^p, b_i) \quad (13)$$

where b_i is a learnable vector designed to capture the collaborative signals from user behavior data. $f(\cdot)$ denotes a function combining the distilled information from the teacher and the information mined from the data. $f(\cdot)$ can be implemented via various ways, e.g., concatenate, MLPs. In our experiments, we find that simple linear combinations suffice to yield satisfactory performance. Such collaborative approach allows our model to leverage the powerful knowledge from the teacher while preserving its capacity to capture collaborative signals.

Remarkably, the distilled student is as efficiency as the original one during the inference. It only requires to leverage LLMs in model training, while directly utilize the well-trained and empowered student model for online service.

4 EXPERIMENTS

Our experiments address the following research questions:

- **RQ1** : Does DLLM2Rec outperform existing distillation strategies? How does the empowered student model perform?
- **RQ2** : What are the impacts of different components of DLLM2Rec on its performance?
- **RQ3** : How do hyperparameters influence DLLM2Rec?

Table 4: Statistics of the datasets.

Datasets	Games	MovieLens	Toys
#Users	55,223	69,878	19,412
#Items	17,408	10,681	11,924
#Interactions	497,577	1,320,000	167,597
Density	0.05176%	0.1769%	0.07241%

4.1 Experiment Settings

4.1.1 Datasets Three conventional datasets: *Amazon Video Games*, *MovieLens-10M*, and *Amazon Toys and Games* were utilized in our experiments. These datasets include user behavior sequences and item content. For fair comparisons, we closely adhered to the pre-processing methods used in recent work [3, 69]. We organized the interaction sequences in ascending order of timestamps to partition each dataset into training, validation, and testing sets with ratios of 8:1:1. Given that MovieLens-10M contains an excessive number of sequences, which could not be processed by LLM-based recommenders, we sampled 100,000 sequences for training and 10,000 for testing, the sampling strategy also adopted by [3]. The dataset statistics are presented in Table 4.

4.1.2 Baselines The following strategies are compared:

Knowledge Distillation Strategies for Recommendation: **Hint** [2], **HTD** [27] are two representative methods that distill knowledge through teacher embeddings; **RD** [53], **CD** [30], **RRD** [26], **DCD** [31] and **UnKD** [8] distill information from the teachers’ ranking lists. Readers may refer to the related work for more details about these strategies.

LLM-enhanced Recommendation Methods: We selected **KAR** [64] and **LLM-CF** [50] for comparisons as they are open-sourced, closely related, and state-of-the-art. **KAR** leverages LLMs as knowledge base to enhance the profile of users and items, while **LLM-CF** uses LLMs to generate a base of chains of thought, which are further retrieved to enhance sequential models.

For fair comparisons, we integrated these methods into three representative sequential models: **GRU4Rec** [18], **SASRec** [28], and

DROS [69], which are either well-known or state-of-the-art. For the teacher model, we consistently used the LLM-based model BIGRec [3], due to its availability as an open-source tool and its demonstrated effectiveness.

4.1.3 Evaluation Metrics We employed two widely-used metrics HR@K and NDCG@K to evaluate performance. Here we simply set K to 20 as recent work [17], and observed similar results with other choices of K.

4.1.4 Implementation Details All methods are implemented with PyTorch and run on 4 Nvidia A800 GPUs. We set $\beta = 1.0$, $\gamma_p = 0.3$, $\gamma_c = 0.5$, $\gamma_o = 0.1$ across all datasets, as these settings were found sufficient to generate good performance, although fine-tuning could further enhance model performance. The influence of these hyperparameters on model performance is also presented in Figure 2. Adam [29] was used as the optimizer with a tuned learning rate of 0.001, a batch size of 256, and weight decay tuned in $\{1e-4, 1e-5, 1e-6, 1e-7, 0\}$, λ_d in $\{0.1, 0.2, \dots, 1.0\}$. We set the embedding size to 64 and the dropout ratio to 0.1.

For all compared methods, we closely followed the settings suggested by their original papers. We also finely tuned their hyperparameters to ensure their optimum. Specifically, for BIGRec, we implemented it with LLaMA2 [55] as suggested by the authors.

4.2 Performance Comparison (RQ1)

The overall experimental results are presented in Table 5.

Comparing with students. The improvements brought by DLLM2Rec are impressive, achieving an average improvement of 47.97% over the original students across three datasets and two metrics. Furthermore, these improvements are consistent under all conditions. These results clearly validate the effectiveness of DLLM2Rec in distilling useful knowledge from the teacher to enhance the student models.

Comparing with existing KDs. DLLM2Rec consistently outperformed all KD baselines across all datasets and metrics. This result clearly validates the effectiveness of DLLM2Rec, with leveraging reliable and student-friendly distillation strategies. We also observed that some KD methods, *e.g.*, Hint and HTD, showed a negative impact on recommendation performance compared to the original student model. This could be attributed to the large semantic gap between the teacher and student models. Additionally, some advanced KD methods like UnKD, DCD, HTD, and RRD may be inferior to the basic RD in some scenarios. This could be attributed to these advanced KD methods adopting more complex distillation strategies, increasing the difficulty for the student to digest the knowledge.

Comparing with existing LLM-enhanced methods. DLLM2Rec consistently outperformed KAR and LLM-CF. This result validates the effectiveness of our distillation paradigm. Compared with KAR and LLM-CF, our distillation strategy effectively leverages the powerful recommendation capabilities of LLMs and directly transfers these merits to the student models. Compared with the Chain of Thought (CoT) utilized by LLM-CF, our distillation strategy directly utilizes the teacher’s embeddings and recommendation results, which could be more easily digested by the student models.

Comparing with the teacher. Table 6 shows the performance and efficiency comparison of the student model empowered by

our DLLM2Rec with the teacher model BIGRec. To our surprise, the empowered lightweight student can even surpass the complex teacher model using LLMs. This result can be attributed to our design — we target letting the student digest the knowledge from the teacher while maintaining its own capacity to capture collaborative signals. Additionally, considering the inference efficiency, the empowered student still maintains low inference latency, while the BIGRec requires an unacceptably long inference time. This result validates that our DLLM2Rec can indeed address a crucial problem—maintaining excellent performance akin to LLM-based recommenders while ensuring low inference latency.

4.3 Ablation Study (RQ2)

We conducted an ablation study on different datasets to study the contributions of each component of DLLM2Rec. For the importance-aware ranking distillation, we removed the entire component (*w/o all_r*), position-aware weights w_{si}^p , confidence-aware weights w_{si}^c , and consistency-aware weights w_{si}^o , respectively. The results are presented in Table 7. For the collaborative embedding distillation, we removed the entire component (*w/o all_e*), the offset term, respectively, and tested the performance when replacing this module with existing embedding distillation strategies including Hint and HTD. The results are presented in Table 8.

As can be seen, both distillation components are important — removing the importance-aware ranking distillation or the collaborative embedding distillation would result in performance drops. Delving deeper into the ranking distillation, we observe that developing the confidence-aware and consistency-aware weights are indeed helpful. For embedding distillation, we observe that the developed offset term is also important. More interestingly, by replacing the entire embedding distillation strategy with Hint and HTD, we observed quite poor performance. This could be attributed to the large semantic gap between the teacher and student models. Blindly aligning the embeddings may harm the model’s semantic space. It could be even worse than directly inheriting the projected space from the teacher.

4.4 Hyperparameter Sensitivity (RQ3)

Figure 2 illustrates performance of DLLM2Rec with different hyperparameters ($\lambda_d, \gamma_p, \gamma_c, \gamma_o$). While we observed some fluctuations, the general trend is that the model’s performance would increase at the beginning and then drop as these parameters increase. This result validates the effectiveness of the corresponding components that each hyperparameter controls. But over-emphasizing one component would incur performance drops as it relatively declines the contribution from others. Finely tuning these hyperparameters for best balance could achieve optimal performance.

5 RELATED WORK

5.1 LLMs for Recommendation System

There are primarily two approaches to utilizing LLMs in RS: LLMs directly as recommenders [3, 4, 33, 71] and LLMs enhancing conventional recommenders [44, 64, 68].

Table 5: Performance comparisons of DLLM2Rec with existing KD methods and LLM-enhanced strategies. The best performance is bold while the runner-up is underlined. *Gain.S* denotes the improvement of DLLM2Rec over the student; while *Gain.B* denotes the improvement of DLLM2Rec over the best baseline.

Backbone	Model	Games		MovieLens		Toys	
		HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
GRU4Rec	Teacher	0.0532	0.0341	0.0541	0.0370	0.0420	0.0207
	+None	0.0305	0.0150	0.0608	0.0236	0.0172	0.0081
	+Hint	0.0284	0.0120	0.0646	0.0240	0.0128	0.0058
	+HTD	0.0299	0.0128	0.0578	0.0229	0.0155	0.0062
	+RD	0.0398	0.0177	0.0667	0.0254	0.0157	0.0076
	+CD	0.0306	0.0149	0.0699	0.0256	0.0126	0.0052
	+RRD	0.0359	0.0163	0.0657	0.0243	0.0215	0.0097
	+DCD	0.0427	0.0190	0.0666	0.0263	0.0262	0.0114
	+UnKD	0.0370	0.0170	0.0607	0.0226	0.0235	0.0114
	KAR	0.0307	0.0149	0.0603	0.0229	0.0184	0.0079
	LLM-CF	0.0393	0.0174	0.0677	0.0246	0.0132	0.0058
	+DLLM2Rec	0.0446	0.0205	0.0815	0.0308	0.0281	0.0118
	<i>Gain.S</i>	+46.17%	+36.94%	+34.05%	+30.43%	+63.88%	+42.18%
	<i>Gain.B</i>	+4.56%	+7.64%	+16.60%	+16.80%	+7.40%	+1.27%
SASRec	+None	0.0346	0.0190	0.0626	0.0228	0.0207	0.0130
	+Hint	0.0358	0.0151	0.0576	0.0216	0.0242	0.0103
	+HTD	0.0343	0.0152	0.0569	0.0214	0.0209	0.0097
	+RD	0.0513	0.0225	0.0778	0.0310	0.0397	0.0164
	+CD	0.0396	0.0231	0.0712	0.0265	0.0232	0.0151
	+RRD	0.0479	0.0202	0.0633	0.0244	0.0325	0.0158
	+DCD	0.0455	0.0211	0.0723	0.0275	0.0375	0.0175
	+UnKD	0.0447	0.0219	0.0667	0.0247	0.0335	0.0174
	KAR	0.0381	0.0198	0.0565	0.0221	0.0215	0.0131
	LLM-CF	0.0559	0.0251	0.0837	0.0295	0.0335	0.0152
	+DLLM2Rec	0.0600	0.0262	0.0840	0.0323	0.0409	0.0177
	<i>Gain.S</i>	+73.55%	+38.25%	+34.19%	+41.91%	+97.68%	+36.38%
	<i>Gain.B</i>	+7.36%	+4.40%	+0.36%	+4.34%	+3.02%	+1.19%
DROS	+None	0.0473	0.0267	0.0852	0.0363	0.0231	0.0144
	+Hint	0.0531	0.0240	0.0791	0.0306	0.0302	0.0135
	+HTD	0.0489	0.0238	0.0722	0.0289	0.0275	0.0137
	+RD	0.0585	0.0310	0.0950	0.0383	0.0424	0.0220
	+CD	0.0474	0.0270	0.0802	0.0336	0.0238	0.0156
	+RRD	0.0590	0.0293	0.0788	0.0338	0.0424	0.0212
	+DCD	0.0531	0.0273	0.0821	0.0348	0.0432	0.0211
	+UnKD	0.0448	0.0209	0.0728	0.0297	0.0375	0.0195
	KAR	0.0586	0.0318	0.0859	0.0352	0.0255	0.0156
	LLM-CF	0.0635	0.0293	0.0963	0.0351	0.0385	0.0178
	+DLLM2Rec	0.0751	0.0331	0.1063	0.0437	0.0463	0.0225
	<i>Gain.S</i>	+58.77%	+23.90%	+24.77%	+20.41%	+100.43%	+56.35%
	<i>Gain.B</i>	+18.27%	+4.03%	+10.38%	+14.24%	+7.07%	+2.16%

LLMs as recommenders. Initial efforts explored the zero-shot capabilities of LLMs in recommendation by structuring the recommendation tasks as language prompts [14, 22, 41, 58, 62]. Subsequently, to adapt LLMs to recommendation tasks, instruction-tuning or fine-tuning has been widely adopted, showing promising results [3, 4, 23, 33–36, 48, 56, 72, 73]. This research primarily focuses on how to enhance LLMs to better suit recommendation tasks. For instance, some studies aim to minimize the semantic gap between recommendations and natural language [3, 6, 20, 25, 37, 51, 60, 66, 74, 77], such as the approach taken by BIGRec [3],

which uses a grounding strategy to map LLM descriptions to recommended items. Others focus on improving LLMs’ ability to model long-sequence interactions [15, 38, 75], and some attempt to reduce training overhead [40]. While effective, these methods often suffer from significant inference inefficiency, limiting their practical application. Although some studies have tried to mitigate inference latency through pre-storage techniques [15] or knowledge distillation [61], the gain of [15] is generally modest and [61] is utilized to distill a huge LLMs (e.g., GPT-115B) to a relatively smaller LLMs (e.g., Llama-7B). Even small Llama-7B is hard to deploy in practical.

Table 6: Performance and efficiency comparison of BIGRec and DLLM2Rec on different datasets.

Dataset	Model	HR@20	NDCG@20	Inference time
Games	BIGRec	0.0532	0.0341	2.3×10^4 s
	DLLM2Rec	0.0751	0.0331	1.8s
	<i>Gain</i>	+37.41%	-2.99%	$+1.3 \times 10^6\%$
MovieLens	BIGRec	0.0541	0.0370	1.8×10^4 s
	DLLM2Rec	0.1063	0.0437	1.7s
	<i>Gain</i>	+96.49%	+18.18%	$+1.1 \times 10^6\%$
Toys	BIGRec	0.0420	0.0207	1.1×10^4 s
	DLLM2Rec	0.0463	0.0225	1.6s
	<i>Gain</i>	+10.24%	+8.70%	$+6.8 \times 10^5\%$

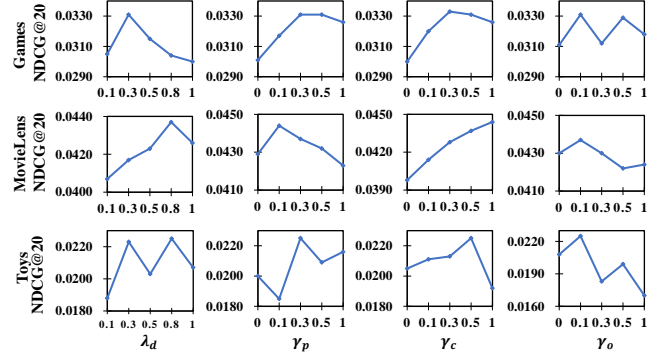
Table 7: Ablation Study on ranking distillation.

Dataset	Model	HR@20	NDCG@20
Games	w/o all_r	0.0661	0.0301
	w/o w_{si}^p	0.0697	0.0301
	w/o w_{si}^c	0.0733	0.0300
	w/o w_{si}^o	0.0568	0.0311
	DLLM2Rec	0.0751	0.0331
MovieLens	w/o all_r	0.0917	0.0364
	w/o w_{si}^p	0.1037	0.0429
	w/o w_{si}^c	0.0986	0.0398
	w/o w_{si}^o	0.1047	0.0430
	DLLM2Rec	0.1063	0.0437
Toys	w/o all_r	0.0386	0.0177
	w/o w_{si}^p	0.0406	0.0200
	w/o w_{si}^c	0.0430	0.0205
	w/o w_{si}^o	0.0445	0.0208
	DLLM2Rec	0.0463	0.0225

Table 8: Ablation Study on embedding distillation.

Dataset	Model	HR@20	NDCG@20
Games	w/o all_e	0.0649	0.0323
	w/o $offset$	0.0700	0.0298
	Hint	0.0563	0.0244
	HTD	0.0568	0.0246
	DLLM2Rec	0.0751	0.0331
MovieLens	w/o all_e	0.0999	0.0420
	w/o $offset$	0.1061	0.0425
	Hint	0.0861	0.0344
	HTD	0.0874	0.0341
	DLLM2Rec	0.1063	0.0437
Toys	w/o all_e	0.0379	0.0194
	w/o $offset$	0.0405	0.0195
	Hint	0.0358	0.0159
	HTD	0.0349	0.0157
	DLLM2Rec	0.0463	0.0225

LLMs enhancing conventional recommenders. Existing methods mainly employ LLMs as supplementary knowledge bases [44, 64, 68] or as encoders for users/items [46, 57, 63] to improve conventional recommenders. For instance, KAR [64] utilizes LLMs as external knowledge bases to better profile users and items within the recommender system. RLMRec [46] encodes user and item

**Figure 2: Sensitivity analysis w.r.t. λ_d , γ_p , γ_c and γ_o .**

profiles into semantic representations and aligns embeddings generated by conventional models with those from LLMs; Wei et al. [63] exploit LLMs to uncover new relationships within graphs. However, compared to direct LLM-based recommenders, these methods do not fully leverage the semantic reasoning capabilities of LLMs for making recommendations. CSRec [68] incorporates the common sense extracted from LLM to alleviate the data sparsity issue.

To capitalize on the reasoning ability of teachers, some efforts have attempted to use chains of thought (CoT) data generated by LLMs to enhance conventional models [50, 61]. However, given the relatively simple architecture of conventional models, it is notably difficult for these systems to assimilate the complex knowledge from CoT data. Furthermore, CoT is typically used as a feature for a sequence, indicating that CoT for a new sequence should be produced during inference, which would be time-consuming. While some approaches attempt to retrieve similar CoT from other user-item pairs [50], such approximations may hurt accuracy.

Differing from these methods, our distillation strategy capitalizes on the superiority of LLMs as recommenders, transferring their exceptional recommendation capabilities to conventional models. Our approach involves direct distillation on embeddings and ranking lists, which are easily assimilated by conventional models without incurring additional computational overhead during inference.

5.2 Sequential Recommendation

Sequential recommendation [7, 9, 32, 49, 65] takes into account the sequence or order of interactions to predict what a user might prefer next. Existing Sequential RS use sequence generation models, such as RNNs or Transformers, to model user interaction sequences. For example, GRU4Rec [18] employs the GRU to handle session-based data, while Caser [52] uses CNN to model interaction data on multiple levels. SASRec [28] introduces attention mechanism to automatically learn the weights of different interaction items, and DROS leverage distributional robust optimization in sequential recommendation and achieves state-of-the-art performance. Most current LLM for RS methods also adopt the setting of sequential recommendation [3, 4]. The readers may refer to the excellent survey [13, 59] for more details.

5.3 Knowledge Distillation in RS

Knowledge distillation (KD) is a promising model compression technique that transfers the knowledge from a large teacher model into the target compact student model [16], and they have been widely

applied in recommendation systems to reduce inference latency. RD [53] treated the top-N ranked items as positive for training a student model; CD [30] utilized soft labels to create positive and negative distillation instances; Soft labels also considered by RRD [26] to create the list-wise distillation loss function; DCD [31] built distillation loss on both user-side and item-side; UnKD [8] addresses popularity bias in distillation. The hidden knowledge among the middle layer of teachers are also considered in some methods. For example, Hint [19] and RRD [26] extracted knowledge of teachers' embedding via Fitnet and expert network; HTD [27] distilled the topological knowledge with the relations in the teacher embedding space. Some researcher also study to distill knowledge from a huge LLMs (e.g., ChatGPT) to a relatively smaller LLMs (e.g., LLaMA-7B) in the recommendation scenarios. Besides model compression, KD is also used to integrate knowledge among different models [54, 67, 76]. For example, some researchers [76] consider to integrate knowledge from multiply pre-trained models into the student. To the best of our knowledge, the study of distilling LLM-based recommenders into conventional recommenders remains untouched.

6 CONCLUSION

This work studies on distilling knowledge from LLM-based recommenders to conventional recommenders. The distillation encounters three challenges including potential unreliable teacher Knowledge, teacher-student capacity gap and semantic space divergence. To tackle this problem, we propose DLLM2Rec with leveraging *importance-aware ranking distillation* and *collaborative embedding distillation* for reliable and student-friendly distillation process. Extensive experiments demonstrate that DLLM2Rec can effectively enhance the performance of three typical lightweight conventional models, with an average improvement of 47.97%, enabling them to keep pace with sophisticated LLM-based models.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
- [2] Romero Adriana, Ballas Nicolas, K Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and Bengio Yoshua. 2015. Fitnets: Hints for thin deep nets. *Proc. ICLR* 2, 3 (2015), 1.
- [3] Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Fuli Feng, Xiangnan He, and Qi Tian. 2023. A bi-step grounding paradigm for large language models in recommendation systems. *arXiv preprint arXiv:2308.08434* (2023).
- [4] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1007–1014.
- [5] 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.
- [6] Yuwei Cao, Nikhil Mehta, Xinyang Yi, Raghunandan Keshavan, Lukasz Heldt, Lichan Hong, Ed H Chi, and Maheswaran Sathiamoorthy. 2024. Aligning Large Language Models with Recommendation Knowledge. *arXiv preprint arXiv:2404.00245* (2024).
- [7] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 378–387.
- [8] Gang Chen, Jiawei Chen, Fuli Feng, Sheng Zhou, and Xiangnan He. 2023. Unbiased knowledge distillation for recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 976–984.
- [9] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 108–116.
- [10] Jang Hyun Cho and Bharath Hariharan. 2019. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4794–4802.
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [12] Shangchen Du, Shan You, Xiaojie Li, Jianlong Wu, Fei Wang, Chen Qian, and Changshui Zhang. 2020. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. *advances in neural information processing systems* 33 (2020), 12345–12355.
- [13] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–42.
- [14] 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).
- [15] Binzong Geng, Zhaoxin Huan, Xiaolu Zhang, Yong He, Liang Zhang, Fajie Yuan, Jun Zhou, and Linjian Mo. 2024. Breaking the Length Barrier: LLM-Enhanced CTR Prediction in Long Textual User Behaviors. *arXiv preprint arXiv:2403.19347* (2024).
- [16] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129, 6 (2021), 1789–1819.
- [17] 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.
- [18] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
- [19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
- [20] Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2023. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*. 1162–1171.
- [21] 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.
- [22] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*. Springer, 364–381.
- [23] Zhiyu Hu, Yang Zhang, Minghao Xiao, Wenjie Wang, Fuli Feng, and Xiangnan He. 2024. Exact and Efficient Unlearning for Large Language Model-based Recommendation. *arXiv:2404.10327 [cs.LG]*
- [24] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. 2022. Knowledge distillation from a stronger teacher. *Advances in Neural Information Processing Systems* 35 (2022), 33716–33727.
- [25] Meng Jiang, Keqin Bao, Jizhi Zhang, Wenjie Wang, Zhengyi Yang, Fuli Feng, and Xiangnan He. 2024. Item-side Fairness of Large Language Model-based Recommendation System. *arXiv preprint arXiv:2402.15215* (2024).
- [26] SeongKu Kang, Junyoung Hwang, Wonbin Kweon, and Hwanjo Yu. 2020. DE-RRD: A knowledge distillation framework for recommender system. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 605–614.
- [27] SeongKu Kang, Junyoung Hwang, Wonbin Kweon, and Hwanjo Yu. 2021. Topology distillation for recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 829–839.
- [28] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
- [29] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [30] Jae-won Lee, Minjin Choi, Jongwuk Lee, and Hyunjun Shim. 2019. Collaborative distillation for top-N recommendation. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 369–378.
- [31] Youngjune Lee and Kee-Eung Kim. 2021. Dual correction strategy for ranking distillation in top-n recommender system. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3186–3190.
- [32] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*. 322–330.

- [33] Lei Li, Yongfeng Zhang, and Li Chen. 2023. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1348–1357.
- [34] Weitao Li, Junkai Li, Weizhi Ma, and Yang Liu. 2024. Citation-Enhanced Generation for LLM-based Chatbots. *arXiv:2402.16063* [cs.CL]
- [35] Xinyi Li, Yongfeng Zhang, and Edward C Malthouse. 2023. Exploring fine-tuning chatgpt for news recommendation. *arXiv preprint arXiv:2311.05850* (2023).
- [36] Zelong Li, Jianchao Ji, Yingqiang Ge, Wenyue Hua, and Yongfeng Zhang. 2024. PAP-REC: Personalized Automatic Prompt for Recommendation Language Model. *arXiv:2402.00284* [cs.IR]
- [37] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2023. Llra: Aligning large language models with sequential recommenders. *arXiv preprint arXiv:2312.02445* (2023).
- [38] Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. 2023. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. *arXiv preprint arXiv:2308.11131* (2023).
- [39] Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2023. A multi-facet paradigm to bridge large language model and recommendation. *arXiv preprint arXiv:2310.06491* (2023).
- [40] Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2024. Data-efficient Fine-tuning for LLM-based Recommendation. *arXiv preprint arXiv:2401.17197* (2024).
- [41] Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2023. A first look at llm-powered generative news recommendation. *arXiv preprint arXiv:2305.06566* (2023).
- [42] Takeshi Ogita, Siegfried M Rump, and Shin'ichi Oishi. 2005. Accurate sum and dot product. *SIAM Journal on Scientific Computing* 26, 6 (2005), 1955–1988.
- [43] Allan Pinkus. 1999. Approximation theory of the MLP model in neural networks. *Acta numerica* 8 (1999), 143–195.
- [44] Jiarui Qin, Weiwen Liu, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. D2K: Turning Historical Data into Retrievable Knowledge for Recommender Systems. *arXiv preprint arXiv:2401.11478* (2024).
- [45] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2024. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2024).
- [46] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Representation learning with large language models for recommendation. *arXiv preprint arXiv:2310.15950* (2023).
- [47] Usha Ruby and Vamsidhar Yendapalli. 2020. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng* 9, 10 (2020).
- [48] Wentao Shi, Xiangnan He, Yang Zhang, Chongming Gao, Xinyue Li, Jizhi Zhang, Qifan Wang, and Fuli Feng. 2024. Enhancing Long-Term Recommendation with Bi-level Learnable Large Language Model Planning. *arXiv preprint arXiv:2403.00843* (2024).
- [49] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.
- [50] Zhongxiang Sun, Zihua Si, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, and Jun Xu. 2024. Large Language Models Enhanced Collaborative Filtering. *arXiv preprint arXiv:2403.17688* (2024).
- [51] Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. 2024. Towards LLM-RecSys Alignment with Textual ID Learning. *arXiv:2403.19021* [cs.IR]
- [52] Jiayi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.
- [53] Jiayi Tang and Ke Wang. 2018. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2289–2298.
- [54] Ye Tao, Ying Li, Su Zhang, Zhirong Hou, and Zhonghai Wu. 2022. Revisiting graph based social recommendation: A distillation enhanced social graph network. In *Proceedings of the ACM Web Conference 2022*. 2830–2838.
- [55] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shrutu Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
- [56] Hangyu Wang, Jianghao Lin, Bo Chen, Yang Yang, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards Efficient and Effective Unlearning of Large Language Models for Recommendation. *arXiv:2403.03536* [cs.IR]
- [57] Hangyu Wang, Jianghao Lin, Xiangyang Li, Bo Chen, Chenxu Zhu, Ruiming Tang, Weinan Zhang, and Yong Yu. 2023. ALT: Towards Fine-grained Alignment between Language and CTR Models for Click-Through Rate Prediction. *arXiv preprint arXiv:2310.19453* (2023).
- [58] Lei Wang and Ee-Peng Lim. 2023. Zero-shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153* (2023).
- [59] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830* (2019).
- [60] Yidan Wang, Zhaochun Ren, Weiwei Sun, Jiyuan Yang, Zhixiang Liang, Xin Chen, Ruobing Xie, Su Yan, Xu Zhang, Pengjie Ren, et al. 2024. Enhanced Generative Recommendation via Content and Collaboration Integration. *arXiv preprint arXiv:2403.18480* (2024).
- [61] Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. 2024. Can Small Language Models be Good Reasoners for Sequential Recommendation? *arXiv preprint arXiv:2403.04260* (2024).
- [62] Zhefan Wang, Weizhi Ma, and Min Zhang. 2024. To Recommend or Not: Recommendation Identification in Conversations with Pre-trained Language Models. *arXiv:2403.18628* [cs.IR]
- [63] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 806–815.
- [64] Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards open-world recommendation with knowledge augmentation from large language models. *arXiv preprint arXiv:2306.10933* (2023).
- [65] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.
- [66] Wentao Xu, Qianqian Xie, Shuo Yang, Jiangxia Cao, and Shuchao Pang. 2024. Enhancing Content-based Recommendation via Large Language Model. *arXiv preprint arXiv:2404.00236* (2024).
- [67] Zixuan Xu, Penghui Wei, Weimin Zhang, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. Ukcd: Debiasing conversion rate estimation via uncertainty-regularized knowledge distillation. In *Proceedings of the ACM Web Conference 2022*. 2078–2087.
- [68] Shenghao Yang, Weizhi Ma, Peijie Sun, Min Zhang, Qingyao Ai, Yiqun Liu, and Mingchen Cai. 2024. Common Sense Enhanced Knowledge-based Recommendation with Large Language Model. *arXiv preprint arXiv:2403.18325* (2024).
- [69] 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 *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 331–340.
- [70] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1285–1294.
- [71] Jizhi Zhang, Keqin Bao, Wenjie Wang, Yang Zhang, Wentao Shi, Wanhong Xu, Fuli Feng, and Tat-Seng Chua. 2024. Prospect Personalized Recommendation on Large Language Model-based Agent Platform. *arXiv:2402.18240* [cs.IR]
- [72] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001* (2023).
- [73] Wenlin Zhang, Xiangyang Li, Yuhao Wang, Kuicai Dong, Yichao Wang, Xinyi Dai, Xiangyu Zhao, Huifeng Guo, Ruiming Tang, et al. 2024. Tired of Plugins? Large Language Models Can Be End-To-End Recommenders. *arXiv preprint arXiv:2404.00702* (2024).
- [74] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Adapting large language models by integrating collaborative semantics for recommendation. *arXiv preprint arXiv:2311.09049* (2023).
- [75] Zhi Zheng, Wenshuo Chao, Zhaopeng Qiu, Hengshu Zhu, and Hui Xiong. 2024. Harnessing Large Language Models for Text-Rich Sequential Recommendation. *arXiv preprint arXiv:2403.13325* (2024).
- [76] Jieming Zhu, Jinyang Liu, Weiqi Li, Jincai Lai, Xiuqiang He, Liang Chen, and Zibin Zheng. 2020. Ensembled CTR prediction via knowledge distillation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2941–2958.
- [77] Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. 2023. Collaborative large language model for recommender systems. *arXiv preprint arXiv:2311.01343* (2023).

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009