C2KD: Cross-layer and Cross-head Knowledge Distillation for Small Language Model-based Recommendations

Xiao Chen[♡♠*] , Changyi Ma[♠*†], Wenqi Fan[♡], Zhaoxiang Zhang[♣◇†], Qing Li[♡†]
[♡]The Hong Kong Polytechnic University
[♠]Center for Artificial Intelligence and Robotics, HKISI-CAS
[♣] Institute of Automation, CAS [◇] University of Chinese Academy of Sciences

Abstract

Sequential recommenders predict users' next interactions based on historical behavior and are essential in modern recommendation systems. While Large Language Models (LLMs) show promise, their size and high inference costs limit deployment on resource-constrained devices. Small Language Models (SLMs) provide a more efficient alternative for edge devices, but bridging the recommendation performance gap between LLMs and SLMs remains challenging. Typical approaches like supervised fine-tuning or vanilla knowledge distillation (KD) often lead to suboptimal performance or even negative transfer. Our motivational experiments reveal key issues with vanilla KD methods: feature imitation suffers from redundancy and uneven recommendation ability across layers, while prediction mimicking faces conflicts caused by differing weight distributions of prediction heads. To address these challenges, we propose a simple yet effective framework, C2KD, to transfer task-relevant knowledge from two complementary dimensions. Specifically, our method incorporates: (1) cross-layer feature imitation, which uses a dynamic router to select the most relevant teacher layers and assimilate task-relevant knowledge from the teacher's late layers, allowing the student to concentrate on the teacher's specialized knowledge; and (2) cross-head logit distillation, which maps the intermediate features of the student to the teacher's output head, thereby minimizing prediction discrepancies between the teacher and the student. Extensive experiments across diverse model families demonstrate that our approach enables 1B-parameter SLMs to achieve competitive performance compared to LLMs (e.g., Llama3-8B), offering a practical solution for real-world on-device sequential recommendations.

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020) have demonstrated great potential in sequential recommendation tasks (Geng et al., 2022; Bao et al., 2023b), leveraging their extensive world knowledge and contextual reasoning. Among them, generation-based LLM recommenders (Li et al., 2023c; Zhang et al., 2023; Li et al., 2023b) are an important branch, which usually inherits the autoregressive architecture of LLMs, transform user behaviors into language prompts, and employ Low-Rank Adaptation (LoRA) (Hu et al., 2021) to repurpose pretrained LLMs for recommendation tasks. However, their large model size and high computational costs hinder deployment in resource-constrained environments. For example, the representative recommender LLARA (Liao et al., 2024) has over 7 billion (7B) parameters and takes 3.8 seconds to process each user sequence, highlighting the need for more efficient alternatives.

Small Language Models (SLMs)[1] have recently garnered attention for achieving comparable performance across various tasks with smaller model size (Wang et al., 2024a; Bellagente et al., 2024; Hu et al., 2024), offering high efficiency and lower deployment costs. These strengths make SLMs well-suited for real-world on-device applications, such as personalized e-commerce recommendations. Despite their potential, the application of pretrained SLMs to sequential recommendation tasks remains largely underexplored, especially in their ability to achieve performance comparable to LLMs.

To address this, we first conduct preliminary studies across various popular SLMs, as shown in Figure 1. The results reveal that directly applying supervised fine-tuning (SFT) to SLMs on recommendation datasets results in significantly worse performance compared to LLMs, implying that

[*] Equal contribution.

[†] Corresponding authors: Changyi Ma (changyi.ma@cair-cas.org.hk), Zhaoxiang Zhang (zhaoxiang.zhang@ia.ac.cn), Qing Li (qing-prof.li@polyu.edu.hk).

[1] Following (Lu et al., 2024; Zhao et al., 2023), we define SLMs as models with fewer than 5B parameters.

Findings of the Association for Computational Linguistics: ACL 2025, pages 17827–17838
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

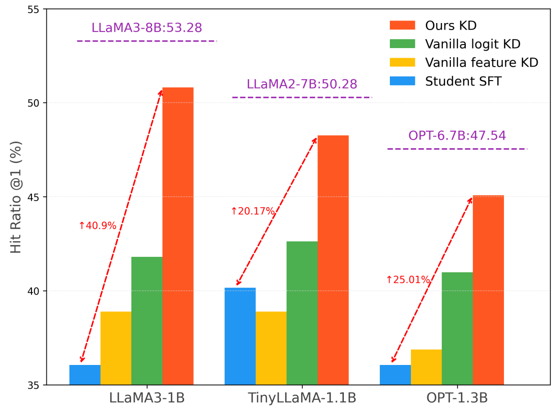


Figure 1: Performance comparison of different methods in fine-tuning SLMs on the LastFM dataset (Cantador et al., 2011). We follow the evaluation protocol in (Liao et al., 2024). The purple dashed line indicates teacher model performance. Our approach significantly outperforms vanilla KD methods.

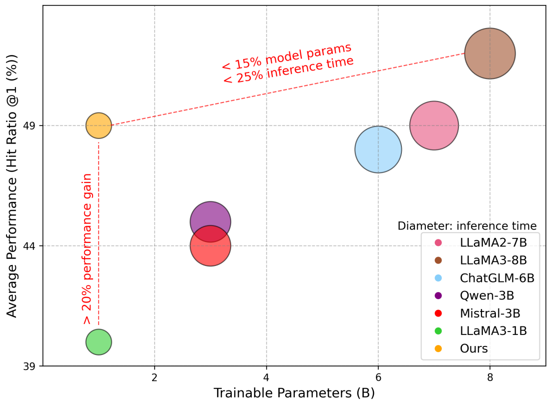


Figure 2: Comparison of model parameters, inference time, and recommendation performance. Our method enables Llama3-1B outperform current SLMs and achieve performance comparable to 7-8B models.

SFT alone is inadequate to equip SLMs with the capabilities needed for effective recommendations. To mitigate the performance gap while preserving the efficiency of SLMs, Knowledge Distillation (KD) offers a promising solution by transferring knowledge from large models to smaller models. While our study (Figure 1) shows that vanilla KD methods (Hinton, 2015; Liang et al., 2023), such as prediction mimicking (‘Vanilla logit KD’) and feature imitation (‘Vanilla feature KD’), often fail to effectively close the performance gap between teacher and student models or even result in negative transfer.

To investigate the limitations of vanilla KD methods for SLM-based sequential recommendations, we conduct motivational experiments in Section 2. We identify two key challenges that hinder effective task-relevant knowledge transfer: **First**, traditional prediction mimicking method suffers from distinct output head weight distributions. In this case, even when teacher and student features are similar, head disparities amplify their differences, which leads to contradictory learning and interferes with optimization. **Second**, the final layer of the LLM does not always deliver the best performance, and the contribution of each LLM layer to recommendation abilities varies across different datasets.

In light of these challenges, we hereby propose a simple yet effective KD framework to empower SLM-based recommenders. Given a well-trained LLM as the teacher, our method transfers knowledge to pretrained SLMs by fine-tuning them with LoRA (Hu et al., 2021), updating only a small set of parameters. The task-relevant knowledge transfer is achieved through two complementary perspectives: **(1) Cross-layer feature imitation**, which aligns the teacher’s and student’s intermediate features using a dynamic router, allowing the student to select the most relevant teacher guidance across different layers. Moreover, to focus on task-relevant knowledge, feature imitation is applied only to the teacher’s late layers, with learnable filters extracting recommendation-related features for more efficient knowledge transfer. **(2) Cross-head prediction mimicking**, which utilizes a learnable projection layer with orthogonal constraints to map the student’s intermediate features into the teacher’s prediction head. This process generates cross-head predictions that are then aligned with the teacher’s original outputs, enabling more effective and harmonious knowledge distillation. Notably, our framework is orthogonal to other LLM post-training efficiency techniques, such as quantization and pruning, and can be combined for further improvements in efficiency.

Without bells and whistles, comprehensive experiments demonstrate that our method consistently exhibits superior performance across various recommendation datasets while maintaining high efficiency. For example, on three large-scale recommendation datasets (Cantador et al., 2011; Harper and Konstan, 2015; Kang and McAuley, 2018), our method enables Llama3-1B to achieve performance comparable to Llama3-8B, despite having a model size less than 15% of Llama3-8B and achieving a 4× inference speedup, as shown in Figure 2. Furthermore, our method consistently outperforms existing KD methods, revealing the importance of designing tailored KD methods for building SLM-based sequential recommenders[2].

2 Motivational Study

In LLM-based recommenders, a common approach is to apply LoRA (Hu et al., 2021) on pretrained LLMs to adapt them to recommendation tasks. However, despite its effectiveness, the inference cost and computational burden of LLMs make them impractical for resource-constrained applications. On the other hand, SLMs, with their smaller size, are well-suited for on-device deployment but often suffer from limited performance when directly fine-tuned on recommendation tasks (as shown in Figure 1). To strike a balance between effectiveness and efficiency, Knowledge Distillation (KD) provides an effective solution by transferring knowledge from large teacher models (LLMs) to smaller student models (SLMs). To assess the applicability of KD in building SLM-based sequential recommenders, we revisit vanilla KD methods and reveal their limitations in this section.

Sequential Recommendation. Given a user's historical interactions in chronological order, a LLM-based sequential recommender aims to select the item i_p that is truly preferred by user u from the candidate set $\mathcal{C}_u = \{i_j\}_{j=1}^{N}$, where N is the number of candidates. Following the LLARA framework (Liao et al., 2024), hybrid prompts that combine collaborative signals from ID-based models (e.g., SASRec (Kang and McAuley, 2018)) with item text descriptions are fed into the LLM. Item IDs are mapped to tokens in the LLM space using a trainable projector. The task-specific loss $\mathcal{L}_{\text{task}}$ is defined in the form of causal language modeling:

$$\mathcal{L}_{task} = \sum_{t=1}^{|y|} \log(P_\Theta(y_t|x_u, y_{<t}; \Theta)), \qquad (1)$$

where x_u denotes the user history interaction sequence, y_t is the t-th token in the model prediction, and Θ represents LLM parameters

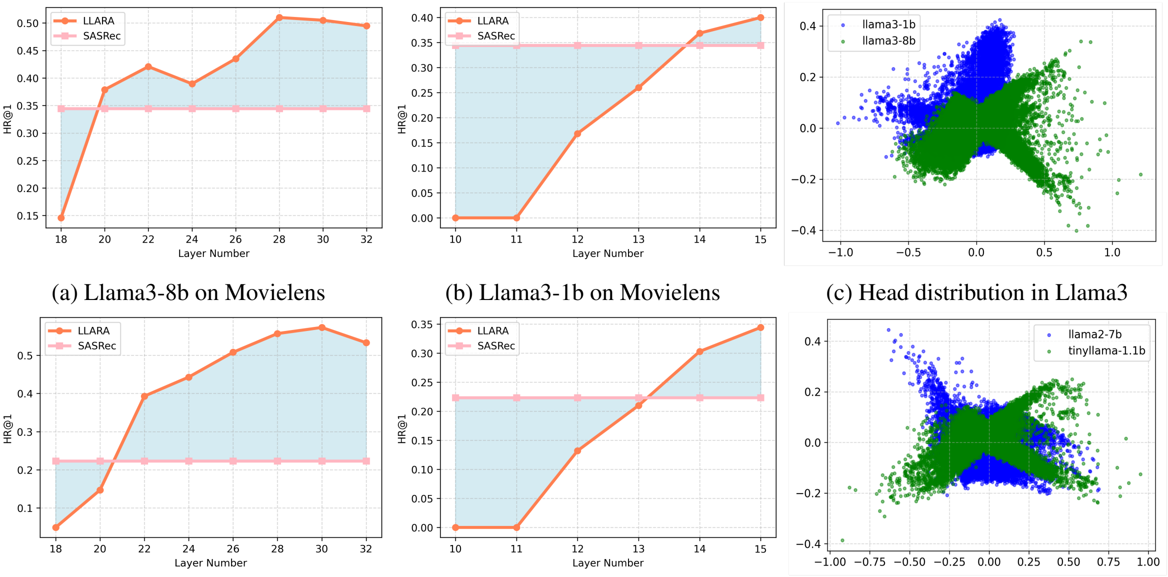
Preliminaries of LLARA We adopt LLARA (Liao et al., 2024) as LLM-based recommendation framework, and the overall architecture is shown in Figure 4. Specifically, the hybrid prompt integrates item textual descriptions with item embeddings derived from traditional recommenders (e.g., SASRec). To align item IDs with the token space of the LLM, a trainable projector is employed. For fair comparisons, we randomly sample negative items from the candidate pool that the user has not interacted with, alongside the positive item as the ground truth. LLM is then repurposed for identifying the true item from these candidate items. It is an end-to-end framework trained with a causal language modeling loss.

Current Knowledge Distillation Frameworks primarily includes two categories: *i) prediction mimicking*, which aligns student prediction distributions with the teacher. It mainly minimizes predict discrepancies between the teacher model and student model via the KL divergence (Van Erven and Harremos, 2014). and *ii) feature imitation*, which aligns hidden representations by minimizing the intermediate feature distance using mean squared error (MSE) as a metric.

However, applying vanilla KD methods to student recommendation models often results in marginal performance gains or even negative transfer (as shown in Fig.1). We suspect these issues arise from two main factors: layer redundancy in the teacher model and the discrepancy between the output heads of the teacher and student models. For **prediction KD**, previous methods (Hinton, 2015; Sun et al., 2024) directly minimize the discrepancy between teacher and student predictions. However, since LLMs and SLMs are pretrained on different datasets, their output heads naturally inherit distinct distribution patterns and lie in different spaces. As a result, directly aligning predictions can be ineffective and may hinder the distillation of useful knowledge. For **feature KD**, previous layer-wise distillation methods (Sun et al., 2019; Liang et al., 2023) use a linear mapping function to align teacher and student feature layers, assuming that performance improves gradually as layers deepen. However, this approach overlooks the potential layer redundancy, where certain intermediate LLM layers contribute less to the recommendation task.

To verify the above conjectures, we conduct preliminary studies on various recommendation datasets, as shown in Fig.3. We employ a layer-wise probing strategy to evaluate the recommendation ability of each decoder layer in LLMs and SLMs. In detail, we halt inference at each decoding layer and use learnable probing heads to map intermediate features to tokens, which are trained with $\mathcal{L}_{\text{task}}$. Additionally, we apply PCA projection to the output head weights of the teacher and student models to visualize the distribution differ-

[2] Codes and models are available at `https://github.com/ShawnChenn/C2KD-LLMRec`



(a) Llama3-8b on Movielens

(b) Llama3-1b on Movielens

(c) Head distribution in Llama3

(d) Llama3-8b on LastFM

(e) Llama3-1b on LastFM

(f) Head distribution in Llama2

Figure 3: The relationship between the number of decoder layers and the recommendation accuracy in the LLM (Llama3-8b) and SLM (Llama3-1b), with the traditional recommender SASRec included as a baseline **(abde)**. Additionally, the output head weight distribution gap between different models **(cf)** is examined to reflect the potential risk of using different heads for prediction mimicking.

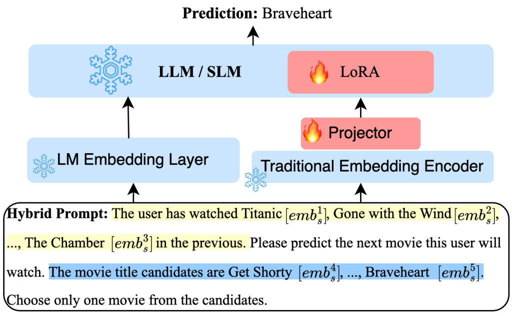

Prediction: Braveheart
LLM / SLM
LoRA
Projector
LM Embedding Layer
Traditional Embedding Encoder
Hybrid Prompt: The user has watched Titanic $[emb_s^1]$, Gone with the Wind $[emb_s^2]$, ..., The Chamber $[emb_s^3]$ in the previous. Please predict the next movie this user will watch. The movie title candidates are Get Shorty $[emb_s^4]$, ..., Braveheart $[emb_s^5]$. Choose only one movie from the candidates.

Figure 4: Overview of the LLARA architecture.

ences between them. Based on these studies, we derive the following insights: *(i) late emergence of recommendation ability.* Sequential recommendation ability typically emerges in the middle to later layers (e.g., after layer 18 in Llama3-8b, layer 12 in Llama3-1b). We speculate earlier layers primarily capture low-level textual cues and lack the context reasoning required for sequential recommendations. *(ii) Intermediate layers outperform final Layer in the LLM.* In Llama3-8b, intermediate layers (e.g., layers 24–30) demonstrate stronger recommendation performance than both the traditional recommender SASRec and the final layer. However, the best-performing layer varies across datasets, suggesting that layer selection in teacher models requires careful designs. While SLM (e.g., Llama3-1b) has a more compact structure, since its intermediate layers do not outperform the final layer. (iii) *Distinct head distribution*. The teacher-student output heads have different distributions across model families. Considering that they are frozen during distillation, even if the teacher and student feature representations are highly similar, the mismatch between their output heads negatively impacts prediction alignment, thereby limiting the effectiveness of KD.

3 Method

In this section, we introduce **C2KD**, a novel **K**nowledge **D**istillation based framework that combines **C**ross-Layer Feature Distillation and **C**ross-Head Prediction Mimicking for SLM-based sequential recommendation. Notably, we keep the pre-trained SLM backbone frozen and leverage LoRA modules to transfer task-specific knowledge from well-trained LLMs to SLMs. This approach significantly reduces training computation costs while avoiding catastrophic forgetting.

3.1 Overview

Fig.5 illustrates the overall workflow of our C2KD, which involves an LLM (e.g., Llama3-8b) as the teacher, denoted as Θ_t, and an SLM (e.g., Llama3-1b) as the student, denoted as Θ_s. Both models have a shared vocabulary space, which is crucial for precise distillation. Our C2KD consists of two

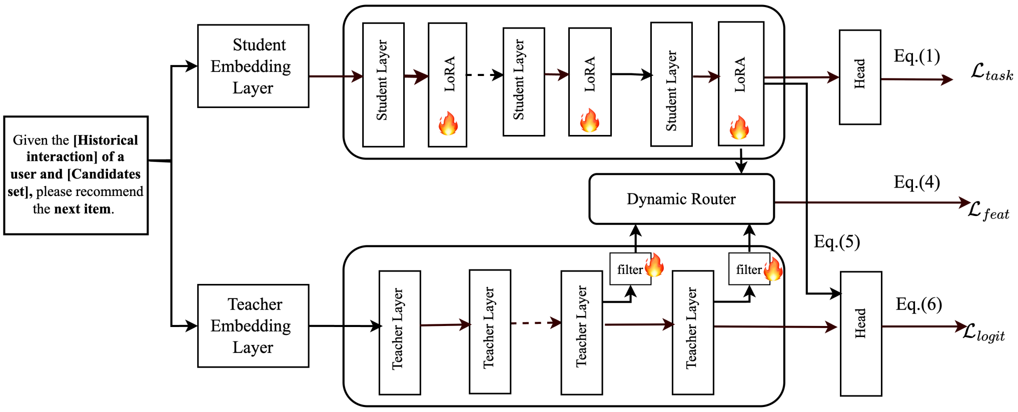


Figure 5: Overview of the proposed C2KD framework.

key components: **a) Cross-layer Feature Distillation**: This component enhances the student model's ability to learn specialized knowledge from hidden representations while addressing the side effects of layer redundancy in the teacher model. **b) Cross-head Prediction Mimicking**: This component introduces a tailored cross-head mechanism to align the predictions between the teacher and student model effectively.

3.2 Cross-layer Feature Distillation

As revealed in Fig.3, the late emergence of recommendation capabilities and uneven layer contributions cast doubt on the effectiveness of the vanilla layer-wise distillation strategy. On one hand, due to the large model capacity gap between teacher and student models, general knowledge may compete with task-specific knowledge during distillation, making feature imitation across all layers suboptimal. On the other hand, the varying performance across layers indicates that a simple linear mapping between teacher and student layers is inadequate for identifying the most relevant teacher layers. These observations underscore the need to rethink what to distill and where to distill to enable more effective knowledge transfer.

First, given the student model's limited capacity, it often struggles to replicate the teacher's extensive knowledge (Durrani et al., 2020). To address this, we focus on distilling task-relevant knowledge from the teacher's late layers, where task-specific information is concentrated. To filter and extract task-specific knowledge, we apply learnable filters to the hidden representations of the teacher model. Let the task-aware filter at layer l be denoted as $g^l(\cdot; W^l)$, where W^l represents its parameters. The filter transforms the original teacher's hidden representations $H_t^l \in \mathbb{R}^{|x| \times d_t}$ into $H_t^l \in \mathbb{R}^{|x| \times d_s}$, where $|x|$ is the sequence length, and d_t and d_s are the dimensions of the teacher's and student's hidden representations, respectively. We apply these filters only to the last M layers of the teacher model, where M is half of the total layers. The learnable filters are jointly optimized as follows:

$$\min_{\mathcal{W}} \sum_{m=1}^{M} -\mathbb{E}_{(x_u,y)}[\mathcal{L}_{\text{task}}(g^m(H^m; W^m))], \quad (2)$$

where $\mathcal{W} = \{W^m\}_{m=1}^{M}$ denote the set of learnable task-aware filter parameters, and $\mathcal{L}_{task}$ represent the recommendation task loss. These filters are well-trained prior to conducting feature distillation.

Second, for a specific student layer, it is crucial to identify the corresponding layer in the teacher model for distillation. Since some intermediate layers of the teacher may outperform the final layer in recommendation tasks, we introduce a dynamic router mechanism that adaptively selects the most relevant teacher layer to guide each student layer. This enables more flexible and effective cross-layer feature distillation. Specifically, we treat each filtered teacher hidden representation $g^m(H_t^m; W^m)$ as a unique feature expert. For the student feature $H_s^n \in \mathbb{R}^{|x| \times d_s}$ at layer n, the dynamic router computes the similarity between the student feature and each filtered teacher feature, and the teacher layer with the highest similarity is automatically selected for feature distillation. This can be formulated as:

$$m^* = \text{argmax}_{m \in \{1,2,\ldots,M\}} \frac{\langle H_s^n, g_t^m(H_t^m; W_t^m)\rangle}{\|H_s^n\| \|g_t^m(H_t^m; W_t^m)\|}. \quad (3)$$

By default, we apply feature imitation only to the student model's last two layers to enhance task knowledge assimilation. Then, the cross-layer feature distillation loss between student layer n and teacher layer m^* can be defined as follows:

$$\mathcal{L}_{\text{feat}}(\Theta_s|\Theta_t) = \text{MSE}\left(g_t^{m^*}(H_t^{m^*}; W_t^{m^*}), H_s^n\right), \tag{4}$$

where the discrepancy between the filtered teacher features and the student features are measured with the mean-squared error.

3.3 Cross-head Prediction Mimicking

As outlined in Fig.3, vanilla prediction mimicking faces challenges due to the distinct output head distribution between teacher and student models, which would prevent the student model from achieving optimal performance.

To address this, we propose a Cross-head distillation method with orthogonal regularization. Instead of solely focusing on output predictions, our approach delivers the student's intermediate features appropriately to the teacher's output head, generating cross-head predictions. This ensures the student effectively mimics the teacher's output distributions. In particular, we introduce a learnable projection layer $W_{\text{proj}} \in \mathbb{R}^{d_s \times d_t}$ to align the dimension of the student's intermediate features with that of the teacher's features.

Moreover, to avoid redundancy or information loss in feature projection (Bansal et al., 2018), we further enforce orthogonality of W_{proj} within the row space and column space, defined as:

$$\mathcal{L}_{\text{orth}} = \|W_{\text{proj}}^\top W_{\text{proj}} - I\| + \|W_{\text{proj}} W_{\text{proj}}^\top - I\|. \tag{5}$$

Then, with the orthogonal constraint in Eq. (5), the aligned student features and teacher features are fed into the same output head of the teacher model. This process is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{logit}}(\Theta_s|\Theta_t) = \mathcal{D}_{\text{pred}}\big(&P_t(y_k|y_{<k}, x_u), \\ &P_s(y_k|y_{<k}, x_u)\big),\end{aligned} \tag{6}$$

where $P_t(y_k|y_{<k}, x_u)$ and $P_s(y_k|y_{<k}, x_u)$ represent the predicted token probabilities of the teacher and student, respectively, given the input sequence x_u. The distance metric $\mathcal{D}_{\text{pred}}$ measures the difference between their output distributions, where we use the Kullback-Leibler (KL) divergence (Van Erven and Harremos, 2014).

3.4 Overall Optimization Objective

The overall training objective is formulated as a weighted sum of the task-specific loss and the distillation loss:

$$\min_{\Theta_s} \mathcal{L}_{\text{task}} + \alpha_1 \mathcal{L}_{\text{logit}} + \alpha_2 \mathcal{L}_{\text{feat}} + \alpha_3 \mathcal{L}_{\text{orth}}, \tag{7}$$

where $\alpha_1, \alpha_2, \alpha_3$ are hyper-parameters that balance various objectives. In this way, we transfer the teacher's complex knowledge to the student model from two complementary perspectives: mimicking the prediction distributions and imitating the feature representations.

4 Experiment

4.1 Experiment Settings

4.1.1 Datasets

We conduct comprehensive experiments on three real-world recommendation datasets: *LastFM* (Cantador et al., 2011), *MovieLens100K* (Harper and Konstan, 2015), and *Steam* (Kang and McAuley, 2018). These datasets consist of user behavior sequences along with item content information. To prepare the data, we sorted the interaction sequences by ascending order of timestamps and split each dataset into training, validation, and testing sets with a ratio of 8:1:1. For fair comparisons, we followed the preprocessing procedures in (Liao et al., 2024; Kong et al., 2024).

4.1.2 Baselines

We compare the following categories of methods:
(i) Traditional Sequential Recommenders: We select **GRU4Rec** (Hidasi, 2015), **SASRec** (Kang and McAuley, 2018), FMLP (Zhou et al., 2022) and **Caser** (Tang and Wang, 2018a) as representative RNN, CNN, all-MLP and attention-based sequential recommenders.
(ii) LLM-based Sequential Recommenders: We consider the following generation-based LLM recommenders: **Llama2** (Touvron et al., 2023), an open-source LLM developed by Meta; **GPT-4** (Achiam et al., 2023) a landmark model released by OpenAI, excelling across diverse tasks; **MoRec** (Yuan et al., 2023), which improves traditional recommenders

Category	Method	LLM	Param	LastFM		MovieLens		Steam	
				HitRatio@1	ValidRatio	HitRatio@1	ValidRatio	HitRatio@1	ValidRatio
Traditional Recommender	GRU4Rec	-	< 1B	0.2616	1.0000	0.3750	1.0000	0.4168	1.0000
	Caser	-		0.2233	1.0000	0.3861	1.0000	0.4368	1.0000
	FMLP	-		0.2541	1.0000	0.3579	1.0000	0.4098	1.0000
	SARSRec	-		0.2233	1.0000	0.3444	1.0000	0.4010	1.0000
LLM-based Recommender	Llama2	Llama2	7B	0.0246	0.3443	0.0421	0.4421	0.0135	0.1635
	GPT-4	-	>175B	0.3770	1.0000	0.2000	0.9895	0.3626	0.9798
	MoRec	RoBERTa	125M	0.1652	1.0000	0.2822	1.0000	0.3911	1.0000
	Tiger	T5	1B	0.3115	1.0000	0.3789	1.0000	0.4132	1.0000
	TALLRec	Llama2	7B	0.4180	0.9836	0.3895	0.9263	0.4637	0.9840
	LLARA	Llama2	7B	0.5080	1.0000	<u>0.4787</u>	0.9895	0.4949	0.9970
Teacher Model	LLARA	Llama3	8B	**0.5328**	1.0000	**0.4947**	1.0000	**0.5413**	1.0000
Student Model	SFT	Llama3	1B	0.3471	0.9918	0.4000	0.9474	0.4425	0.9890
	SeqKD	Llama3	1B	0.4180	1.0000	0.4105	0.9684	0.4471	0.9684
	LogitKD	Llama3	1B	0.4098	1.0000	0.4222	0.9474	0.4376	0.9474
	Hint	Llama3	1B	0.3770	1.0000	0.4315	0.9789	0.4611	0.9831
	TAD	Llama3	1B	0.4262	1.0000	0.4517	0.9789	0.4796	0.9915
	Ours	Llama3	1B	<u>0.5163</u>	1.0000	**0.4947**	0.9894	<u>0.4966</u>	0.9940

Table 1: Comparison with representative methods on the recommendation benchmarks. We compare our method with Traditional Recommenders and LLM-based generative recommenders. The best result is in **bold**, and the second-best result is <u>underlined</u>. *(p-value $\ll$ 0.05)

by incorporating LLM-encoded item modality features, such as text information; **TALLRec** (Bao et al., 2023b), which adopts instruction tuning for LLMs on recommendation datasets; **TIGER** (Rajput et al., 2023), a Generative Retrieval framework for Recommender Systems based on Semantic ID representation of items; **LLARA** (Liao et al., 2024), which integrates LLM embeddings with ID embeddings from traditional recommenders effectively.

(iii) Distillation methods: We select **Hint** (Romero et al., 2014) and **TAD** (Liang et al., 2023) as representative methods that distill knowledge using teacher embeddings; **LogitKD** (Hinton, 2015), which uses output logits from the teacher model as soft labels to guide the student; **SeqKD** (Kim and Rush, 2016), which trains the student with data generated by the teacher. For sequential recommendation, following (Wang et al., 2024b), we use recommendation rationales as the generated data.

4.1.3 Evaluation Metrics

Following prior works on generation-based LLM recommenders (Liao et al., 2024; Kong et al., 2024), for each user sequence, we randomly select 20 non-interacted items to construct the candidate set, including the correct subsequent item. All methods aim to identify the correct item, with performance evaluated using **i) HitRatio@1**. LLM-based recommenders generate a single candidate item via prompting, while traditional models predict the item with the highest probability. Since LLM-based generative approaches may produce hallucinated responses, we introduce an additional metric **ii) Valid ratio@1**, which measures the proportion of valid responses within the candidate set.

4.1.4 Implementation Details

Our method is implemented in PyTorch and runs on 4 NVIDIA A100 GPUs. The training process involves a total of 5 epochs. The teacher model, initialized from a pretrained LLM (e.g., Llama3-8B (Dubey et al., 2024)), is fine-tuned on recommendation datasets following LLARA (Liao et al., 2024). By default, LLARA uses SASRec (Kang and McAuley, 2018) to extract user behavior tokens. During distillation, the student model (e.g., Llama3-1B) is initialized from an SLM and fine-tuned using LoRA to efficiently adapt to the target recommendation tasks. Note that our method requires the teacher and student models to have aligned vocabulary, i.e., identical vocabulary sizes. The hyperparameters are set as $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, and $\alpha_2 = 0.01$ across different datasets. Our method shows a relatively low sensitivity to hyperparameter value selection.

4.2 Performance Comparison

Table 1 showcases the quantitative comparison on three large-scale sequential recommendation datasets. We have several key observations: (1) Compared to traditional recommenders, LLM-based generative recommenders usually due to their extensive world knowledge. Due to scaling laws, larger models with more parameters often achieve better recommendation performance. (2) Our distillation approach substantially narrows the performance gap between the teacher and student model, enabling SLMs to achieve performance comparable to their teacher models. This improvement is attributed to its ability to absorb teacher knowledge from multiple perspectives, resulting in approximately a 28% performance enhancement over supervised finetuning on the student model, i.e., 'SFT'. (3) Vanilla knowledge distillation techniques typically fail to effectively mitigate the significant performance gap between the teacher and student models and may even risk negative transfer.

4.3 Ablation Study

As shown in Table 2, we ablate different modules in our method to evaluate their importance. With different knowledge distillation regularizers (namely 'Ours cross-layer feat' and 'Ours cross-head logit'), the framework demonstrates improved performance compared to using only supervised fine-tuning ('SFT'). The two distillation strategies are complementary when combined ('Ours w/o orthogonal'). Meanwhile, $\mathcal{L}_{\text{orth}}$ consistently facilitates stable improvements in cross-head logit distillation or when using both feature and logit distillation ('Ours').

Method	$\mathcal{L}_{\text{task}}$	$\mathcal{L}_{\text{feat}}$	$\mathcal{L}_{\text{logit}}$	$\mathcal{L}_{\text{orth}}$	Hit Ratio@1	Valid Ratio@1
SFT	✓				0.3471	0.9918
Ours cross-layer feat	✓	✓			0.4344	1.0000
Ours logit	✓		✓		0.4426	1.0000
Ours cross-head logit	✓		✓	✓	0.4672	1.0000
Ours w/o orthogonal	✓	✓	✓		0.5081	1.0000
Ours	✓	✓	✓	✓	0.5163	1.0000

Table 2: Ablation study on the LastFM dataset, where the teacher model is Llama3-8B and the student model is Llama3-1B.

4.4 Effectiveness and Efficiency Analysis

Our proposed distillation framework can be applied in various LLM families, as shown in Table 3. The teacher model can be a pre-trained LLM (e.g., Llama2-7B (Touvron et al., 2023), OPT-6.7B (Zhang et al., 2022)) fine-tuned on recommendation datasets. The student model is a smaller LLM with the same vocabulary, such as TinyLLaMA (Zhang et al., 2024b), OPT-1.3B (Zhang et al., 2022)).

It can be observed that our method consistently mitigates the performance gap between the teacher and student models. We report the time efficiency and parameters of comparative baseline LLARA and our model in Table 4.

Dataset	**Llama2 7B → TinyLlama 1.1B**			**OPT 6.7B → OPT 1.3B**		
	Teacher	**Student**	**Ours**	**Teacher**	**Student**	**Ours**
Lastfm	50.28	40.16	47.54	45.08	39.34	47.54
Movielens	52.53	47.39	49.96	46.67	38.94	45.26
Steam	47.36	43.16	47.36	45.83	36.93	44.94

Table 3: Performance comparison of HitRatio @1 across various models.

Method	LLaRA	Ours-Llama3	Ours-TinyLlama	Ours-OPT
Inf time (s)	3.81	0.84	2.10	2.23
Params (b)	6.79	1.24	1.11	1.30

Table 4: Efficiency comparison between LLaMA and our method on various SLMs in terms of instance-wise inference time (seconds) and total number of model parameters (billion).

5 Related Work

5.1 LLM-based Sequential Recommenders

Sequential recommendation focuses on predicting the next item a user is likely to engage with based on their previous interaction history. Early research primarily relied on modeling user preferences using architectures like RNNs (HidasiB et al., 2015), CNNs (Tang and Wang, 2018a), or Transformers (Kang and McAuley, 2018). With the rise of LLMs, their extensive knowledge and reasoning abilities show great promise for sequential recommendation. Integrating LLMs into recommendation systems typically follows two paradigms: *LLMs as recommenders* (Bao et al., 2023a,b; Liao et al., 2024), where LLMs are repurposed for recommendations via fine-tuning, prompting, or in-context learning; and *LLMs as enhancers* (Xi et al., 2024; Ren et al., 2024; Wang et al., 2023), where LLMs provide feature embeddings or rationales but still rely on conventional relevance calculations, limiting their reasoning potential. In this paper, we adopt the first architecture for both the teacher and student models, and follow (Liao et al., 2024) to integrate collaborative information from traditional models. Though LLM-based recommenders achieve notable progress, their high inference cost and large model size prohibit deployment on edge devices. Recently, SLMs have gained attention for achieving comparable performance on various tasks with significantly smaller model sizes (Wang et al., 2024a; Lu et al., 2024). However, their potential in recommendations remains underexplored.

5.2 Knowledge Distillation in RecSys

Knowledge distillation (KD) compresses models by transferring knowledge from a large teacher model to a smaller student model. For LLM distillation, some works (Kim and Rush, 2016; Gu et al., 2024b,a) train small student models on teacher's generated text data. Other works explore better optimization goals for distillation, such as aligning token-level probability distribution (Muralidharan et al., 2024), hidden features (Sun et al., 2019) and attention matrices (Wang et al., 2020b,a). Early RecSys methods (Tang and Wang, 2018b; Lee et al., 2019; Kang et al., 2020; Lee and Kim, 2021; Chen et al., 2023; Fan et al., 2022) focus on traditional recommenders, utilizing the teacher's top-N items as soft labels or distilling knowledge from embeddings and topological relationships (Hinton, 2015; Kang et al., 2021). With the rise of LLM-based recommender architectures (Li et al., 2023a; Hou et al., 2024), there is growing interest in distilling knowledge from LLMs to lighter models (Wu et al., 2024; Cui et al., 2024). For example, (Wang et al., 2024b) introduces chain-of-thought distillation to transfer reasoning abilities from LLMs (e.g., ChatGPT) into the smaller model Llama2-7B, which remains impractical for resource-limited scenarios. SMLRec(Xu et al., 2024) employs vanilla feature imitation but enforces strict architectural constraints, such as requiring identical hidden dimensions between teacher and student models. Additionally, it does not leverage popular SLMs, limiting its flexibility and adaptability.

5.3 Small Language Models

There has been a growing interest in developing small language models (SLMs) recently, which aim to maintain comparable task performance while significantly improving inference efficiency (Wang et al., 2024a; Lu et al., 2024). These SLMs can generally be categorized into two types: *general-domain SLMs* (Zhang et al., 2024b; Thawakar et al., 2024; Abdin et al., 2024), which are designed to acquire extensive general knowledge and capabilities with compact model size (i.e., less than 5B); and *domain-specific SLMs* (Zhang et al., 2024a; Bolton et al., 2024), which focus on well-defined tasks and expertise pertinent to specific fields (e.g., scientific or biomedical domain). In this paper, we primarily focus on fine-tuning general-domain SLMs using Low-Rank Adapters (LoRA) (Hu et al., 2021), which enable SLMs to be tailored to sequential recommendation tasks while ensuring low computational and memory costs.

6 Conclusion

This paper proposes a novel distillation framework that efficiently transfers task-relevant knowledge from LLMs to SLMs for sequential recommendation. Our motivational study reveals the limitations of vanilla KD methods, including layer redundancy and uneven recommendation ability across LLM layers, along with differing weight distributions in prediction heads. To address these challenges, our method leverages cross-layer feature imitation and cross-head logit distillation, enabling harmonious and effective task-relevant knowledge transfer. Extensive experiments demonstrate that 1B-parameter SLMs can achieve performance comparable to 8B-parameter LLMs, providing a practical and scalable solution for recommendation systems on resource-constrained devices.

Acknowledgments

This research work was supported by the InnoHK program. The research described in this paper has also been partly supported by General Research Funds from the Hong Kong Research Grants Council (project no. PolyU 15207322, 15200023, 15206024, and 15224524), internal research funds from The Hong Kong Polytechnic University (project no. P0042693, P0048625, P0051361, P0052406, and P0052986).

Limitations

Despite the promising results demonstrated by our method, it is important to acknowledge its limitations. One key limitation lies in the empirical nature of our findings, as the motivations are primarily based on observed experimental results. Additionally, due to limited computational resources, we were unable to conduct experiments on larger models, such as Llama2-13B or beyond. This restricts our ability to fully evaluate the scalability and generalizability of the proposed distillation framework on larger-scale models. Future work could address these issues by exploring more diverse model sizes and conducting a deeper theoretical analysis to further enhance our approach.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. 2018. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31.

Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Chong Chen, Fuli Feng, and Qi Tian. 2023a. A bi-step grounding paradigm for large language models in recommendation systems. *arXiv preprint arXiv:2308.08434*.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023b. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1007–1014.

Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, et al. 2024. Stable lm 2 1.6 b technical report. *arXiv preprint arXiv:2402.17834*.

Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, et al. 2024. Biomedlm: A 2.7 b parameter language model trained on biomedical text. *arXiv preprint arXiv:2403.18421*.

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*, 33:1877–1901.

Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). In *Proceedings of the fifth ACM conference on Recommender systems*, pages 387–388.

Xiao Chen, Wenqi Fan, Jingfan Chen, Haochen Liu, Zitao Liu, Zhaoxiang Zhang, and Qing Li. 2023. Fairly adaptive negative sampling for recommendations. In *Proceedings of the ACM Web Conference 2023*, pages 3723–3733.

Yu Cui, Feng Liu, Pengbo Wang, Bohao Wang, Heng Tang, Yi Wan, Jun Wang, and Jiawei Chen. 2024. Distillation matters: empowering sequential recommenders to match the performance of large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 507–517.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. *arXiv preprint arXiv:2010.02695*.

Wenqi Fan, Xiangyu Zhao, Xiao Chen, Jingran Su, Jingtong Gao, Lin Wang, Qidong Liu, Yiqi Wang, Han Xu, Lei Chen, et al. 2022. A comprehensive survey on trustworthy recommender systems. *arXiv preprint arXiv:2209.10117*.

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*, pages 299–315.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024a. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Yuxian Gu, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024b. Miniplm: Knowledge distillation for pre-training language models. *arXiv preprint arXiv:2410.17215*.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

B Hidasi. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Karatzoglou A HidasiB et al. 2015. Session-based recommendationswithrecurrentneuralnetworks. *arXiv preprint arXiv:1511.06939*.

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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*, pages 364–381. Springer.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

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*, pages 605–614.

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*, pages 829–839.

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

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Xiaoyu Kong, Jiancan Wu, An Zhang, Leheng Sheng, Hui Lin, Xiang Wang, and Xiangnan He. 2024. Customizing language models with instance-wise lora for sequential recommendation. *arXiv preprint arXiv:2408.10159*.

Jae-woong Lee, Minjin Choi, Jongwuk Lee, and Hyunjung Shim. 2019. Collaborative distillation for top-n recommendation. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 369–378. IEEE.

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*, pages 3186–3190.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023a. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1258–1267.

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

Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. 2023c. Large language models for generative recommendation: A survey and visionary discussions. *arXiv preprint arXiv:2309.01157*.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, pages 20852–20867. PMLR.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1785–1795.

Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D Lane, and Mengwei Xu. 2024. Small language models: Survey, measurements, and insights. *arXiv preprint arXiv:2409.15790*.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Bhuminand Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Compact language models via pruning and knowledge distillation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3464–3475.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. 2024. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15731–15740.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Jiaxi Tang and Ke Wang. 2018a. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 565–573.

Jiaxi Tang and Ke Wang. 2018b. 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*, pages 2289–2298.

Omkar Thawakar, Ashmal Vayani, Salman Khan, Hisham Cholakal, Rao M Anwer, Michael Felsberg, Tim Baldwin, Eric P Xing, and Fahad Shahbaz Khan. 2024. Mobillama: Towards accurate and lightweight fully transparent gpt. *arXiv preprint arXiv:2402.16840*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tim Van Erven and Peter Harremos. 2014. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820.

Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, et al. 2024a. A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. *arXiv preprint arXiv:2411.03350*.

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*.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2020a. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. *arXiv preprint arXiv:2012.15828*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. 2024b. Can small language models be good reasoners for sequential recommendation? In *Proceedings of the ACM on Web Conference 2024*, pages 3876–3887.

Xuansheng Wu, Huachi Zhou, Yucheng Shi, Wenlin Yao, Xiao Huang, and Ninghao Liu. 2024. Could small language models serve as recommenders? towards data-centric cold-start recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3566–3575.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 12–22.

Wujiang Xu, Qitian Wu, Zujie Liang, Jiaojiao Han, Xuying Ning, Yunxiao Shi, Wenfang Lin, and Yongfeng Zhang. 2024. Slmrec: empowering small language models for sequential recommendation. *arXiv preprint arXiv:2405.17890*.

Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2639–2649.

Dan Zhang, Ziniu Hu, Sining Zhoubian, Zhengxiao Du, Kaiyu Yang, Zihan Wang, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Sciglm: Training scientific language models with self-reflective instruction annotation and tuning. *arXiv preprint arXiv:2401.07950*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024b. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

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

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced mlp is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022*, pages 2388–2399.