Bi-Tuning with Collaborative Information for Controllable LLM-based Sequential Recommendation

Anonymous ACL submission

Abstract

Sequential recommender systems, which leverage historical interactions to deliver targeted recommendations, have been significantly advanced by large language models (LLMs). However, LLM-based generative sequential recommendation often faces two key challenges: the lack of collaborative knowledge and the limited controllability over the generated content. In this paper, we propose a simple Bi-Tuning framework with collaborative information for controllable Large Language Model-based Sequential Recommendation (*Laser*). Specifically, *Bi-Tuning* works through incorporating learnable virtual tokens at both the prefix and suffix of the input text, where the prefix tokens enable the adaptation of LLMs with collaborative information, while the suffix token transforms the LLM output into item/user embeddings for similarity comparison, thereby facilitating controllable recommendations. Furthermore, we introduce an *MoE-based querying transformer* that selectively activates experts to extract relevant information from varying collaborative signals of frozen ID-based recommenders into the prefix, coupled with a multi-task loss function incorporating the MoE load-balancing objective. Finally, a two-phase training strategy is employed to progressively obtain high-quality item and user embeddings through the learnable suffix. Experiments on real-world datasets show that Laser effectively adapts LLMs for sequential recommendation, outperforming state-of-the-art baselines.

1 Introduction

Sequential recommender systems have become essential across various applications, aiming to predict users' future preferences based on their past behaviors. While early works primarily relied on item ID sequences to capture the dynamic nature of user preferences (He and McAuley, 2016; Hidasi et al., 2016; Li et al., 2017; Tang and Wang, 2018), recent studies have incorporated item textual information (e.g., item titles, categories, and brands) based on pre-trained language models (PLMs) to enrich item sequence representations and enhance recommendation performance (Hou et al., 2022; Li et al., 2023).

Recently, the emergence of large language models (LLMs) has triggered a significant revolution in the research community (Lin et al., 2024; Wang et al., 2024a). With the powerful instruction-following capability, LLMs can effectively generate personalized recommendations based on recommendation instructions containing user interaction history and candidate item information (Bao et al., 2023; Hou et al., 2024). However, despite their potential, LLM-based recommendation systems primarily rely on text semantics, inherently overlooking collaborative signals. As a result, for the same user, items with similar textual descriptions may be recommended in a similar manner, even if their user interaction patterns differ significantly (Chen et al., 2023; Zhang et al., 2024). While some efforts, such as using multilayer perceptrons (MLPs) to map collaborative embeddings encoded by traditional ID-based collaborative models into the LLM semantic space (Yang et al., 2023; Zhang et al., 2023b), have been explored, the integration of collaborative knowledge with LLMs still remains an open challenge.

Furthermore, generative sequential recommender systems based on LLMs may suffer from the limited controllability over the generated content (Lu et al., 2024). These models, typically trained to generate recommended items through the next-token prediction loss (Qiu et al., 2023; Kim et al., 2024), may introduce domain-specific formatting errors, such as irrelevant or repeated items. To address these issues, additional alignment strategies, such as auxiliary supervised learning tasks (Zhang et al., 2023a; Zheng et al., 2024) or reinforcement learning tasks (Lu et al., 2024), are of-

ten required, which, however, introduce increased method complexity and computational overhead.

In this paper, we propose a simple Bi-Tuning framework with collaborative information for controllable Large Language Model-based Sequential Recommendation (***Laser***). In ***Bi-Tuning***, we adapt LLMs for sequential recommendation by optimizing learnable virtual tokens inserted at both the prefix and suffix of the input text in a parameter-efficient manner. Specifically, the prefix tokens are responsible for adapting LLMs with collaborative information, while the suffix token transforms the LLM output from the language space to the recommendation space, generating item/user embeddings used for similarity comparison, thereby facilitating controllable next-item recommendation. Furthermore, to integrate collaborative knowledge into the prefix, we propose ***M-Former***, a lightweight Mixture-of-Experts (MoE)-based querying transformer that selectively activates different experts to extract relevant information from varying collaborative signals of frozen ID-based recommenders into the prefix. Additionally, to ensure balanced expert utilization in M-Former, we introduce a multi-task loss that simultaneously optimizes both the recommendation and load-balancing objectives. Finally, we adopt a two-phase training strategy to progressively obtaining high-quality item and user embeddings through the learnable suffix. Experimental results on real-world datasets across various domains show that our method significantly outperforms state-of-the-art baselines. In summary, our main contributions are as follows:

1) We propose Laser, a simple but effective Bi-Tuning (through learnable prefix and suffix) framework with collaborative information for controllable LLM-based sequential recommendation.

2) We introduce M-Former that selectively activates different experts to extract relevant information from varying collaborative signals of frozen ID-based recommenders into the prefix, paired with a multi-task loss considering the load-balancing objective. Additionally, a two-phase training strategy is employed to progressively obtain high-quality item and user embeddings through the learnable suffix.

3) Extensive experiments on real-world datasets show that Laser[1] effectively adapts LLMs for sequential recommendation, significantly outperforming state-of-the-art baselines.

[1]Our code is available at: https://anonymous.4open.science/r/Laser-main-B1A5.

2 Related Work

2.1 Sequential Recommendation

Sequential recommendation infers users' preferences from past interactions ordered by timestamps, with traditional methods typically representing items using unique IDs. To effectively capture user preferences based on the IDs, a variety of methods have been employed. For instance, GRU4Rec (Chung et al., 2014) models sequential patterns with GRUs, while Caser (Tang and Wang, 2018) embeds the sequence of recent items into an "image" and learn sequential patterns using convolutional filters. Methods like SR-GNN (Wu et al., 2019), GCE-GNN (Wang et al., 2020), and SURGE (Chang et al., 2021) capture long-term user preferences through multi-layer message passing, and self-attention models have also been widely adopted (Kang and McAuley, 2018; Sun et al., 2019; Li et al., 2020). Although these ID-based methods show promise, they fail to incorporate semantic information from item descriptions, resulting in suboptimal performance. Recently, researchers have explored using PLMs to encode item textual information, enriching item sequence representations and improving recommendation performance (Hou et al., 2022; Li et al., 2023), though most efforts have focused on smaller language models.

2.2 LLMs in Recommender Systems

Due to the powerful instruction-following capability of LLMs, an increasing number of works have attempted to express users' past interactions along with candidate item information as natural language instructions, enabling LLMs to generate tailored recommendations (Bao et al., 2023; Zhang et al., 2023a; Hou et al., 2024). However, LLM-based generative sequential recommender systems often face two challenges. First, they primarily rely on text semantics, representing users and items as textual tokens rather than leveraging explicit interaction patterns. As a result, they inherently overlook collaborative signals embedded in user-item co-occurrences, leading to suboptimal performance (Chen et al., 2023; Zhang et al., 2024). While efforts have been made to address this, such as mapping collaborative embeddings through MLPs (Chen et al., 2023; Zhang et al., 2024) or incorporating collaborative information into the LLM

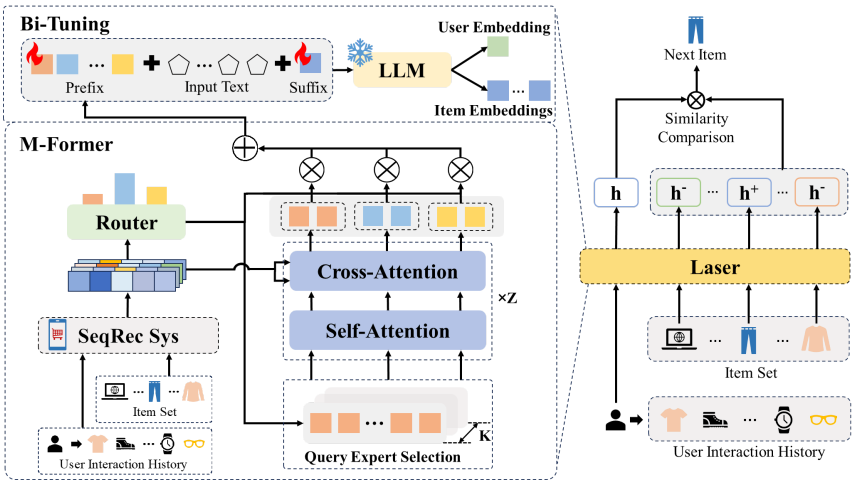

Figure 1: The overview of our proposed Laser.

attention weight calculation (Wang et al., 2024b), integrating collaborative knowledge with LLMs remains an open challenge. Second, generative recommendation systems often suffer from limited controllability over the generated content (Lu et al., 2024). Previous works have introduced additional training tasks to regularize the LLM output, such as auxiliary supervised learning tasks (Zhang et al., 2023a; Zheng et al., 2024) or reinforcement learning tasks (Lu et al., 2024). While these methods show promising results, they largely increase the model complexity and computational overhead. In this paper, we propose Laser, a simple Bi-Tuning framework with collaborative information for controllable large language model-based sequential recommendation, which adapts LLMs for sequential recommendation by optimizing learnable virtual tokens at both the prefix and suffix in a parameter-efficient manner.

3 Preliminaries

In sequential recommendation, $\mathcal{U}$ denotes the set of users and $\mathcal{I}$ represents the set of items. Each user $u \in \mathcal{U}$ has a temporally ordered sequence of interacted items $S_u = \{i_1, i_2, \ldots, i_N\}$, where N is the sequence length and $i \in \mathcal{I}$. The goal is to predict the next item i_{N+1}. In this work, we transform each user's interaction history S_u and the information of each item i into a natural language instruction using a pre-defined template (detailed in Section 4.1). The instruction represented as $T = \{t_1, t_2, \ldots, t_W\}$, where W is the text length, prompts LLMs to generate user/item embeddings for similarity comparison and controllable next-item recommendation.

4 Method

In this section, we present our proposed Laser. First, we explain how the Bi-Tuning framework adapts LLM for sequential recommendation through the learnable prefix and suffix. Next, we describe how the M-Former activates different experts to extract relevant information from different collaborative signals into the prefix. Then, we introduce the multi-task loss function that balances the recommendation and load-balancing objectives. Finally, a two-phase training strategy is designed to progressively obtain high-quality item/user embeddings.

4.1 Bi-Tuning for LLM-based Sequential Recommendation

Inspired by previous works (Li and Liang, 2021; Lester et al., 2021), we propose an innovative way to adapt LLMs to the sequential recommendation task by adding learnable virtual tokens to both the prefix and suffix of the input text. Specifically, given the input instruction $T = \{t_1, t_2, ..., t_W\}$, it is expanded with learnable prefix and suffix tokens:

$$\tilde{T} = \{\underbrace{p_1, p_2, ..., p_L}_{\text{prefix}}, \underbrace{t_1, t_2, ..., t_W}_{\text{instruction}}, \underbrace{s}_{\text{suffix}}\}, \quad (1)$$

where $P = \{p_1, p_2, ..., p_L\}$ represents the *prefix* containing L prepended virtual tokens, and s denotes the *suffix*, consisting of a single appended virtual token. The virtual tokens serve as placeholders for LLM fine-tuning, while the LLM's parameters remain frozen. Specifically, the prefix is responsible for adapting LLMs with collaborative knowledge, which we will detail in Section 4.2 using the proposed *M-Former*. The suffix token

is designed to capture the interaction history of a user u or the description of a single item i in the instruction T based on the following template:

> You are an intelligent recommendation assistant. Please summarize the user's characteristics into *a single token* based on the interaction history. In chronological order, the user has interacted with the following items:
> >> 1. Kaytee Aspen Bedding Bag (brand: Kaytee, category: Kaytee)
> >> 2. ...

Specifically, we treat a single item as a special case of user interaction history that contains only the item itself. In this way, a unified template can process both user and item information, minimizing its impact on the performance of LLMs (Lester et al., 2021; Liu et al., 2022).

Then, the output embedding of the suffix token, based on the designed template, is used as the user embedding $\mathbf{h}_u \in \mathbb{R}^d$ or item embedding $\mathbf{h}_i \in \mathbb{R}^d$ for similarity comparison and next-item prediction, where d represents the LLM hidden size:

$$s(u,i) = \cos(\mathbf{h}_u, \mathbf{h}_i) = \frac{\mathbf{h}_u^\top \mathbf{h}_i}{\|\mathbf{h}_u\| \, \|\mathbf{h}_i\|}, \tag{2}$$

$$\hat{i} = \operatorname{argmax}_{i \in \mathcal{I}} \left(s(u,i) \right). \tag{3}$$

Through this way, we mitigate the uncontrollability issue in generative recommendation.

4.2 M-Former for Collaborative Information Integration

To effectively integrate collaborative information into LLMs for more accurate recommendations, we propose M-Former, a lightweight MoE-based querying transformer. As shown in Figure 1, M-Former selectively activates different query experts to extract relevant collaborative information from frozen ID-based recommenders. The experts carrying collaborative knowledge are then aggregated to form the prefix, which better adapts the LLMs with such collaborative information.

Specifically, to provide LLMs with collaborative knowledge, following previous works (Yang et al., 2023; Zhang et al., 2023b), we employ a frozen ID-based sequential recommender to generate collaborative embeddings $\mathbf{C} \in \mathbb{R}^{N \times d_c}$ based on the input item ID sequence, where N denotes the sequence length and d_c is the hidden size of the recommender. The input to the ID-based recommender aligns with the input to the LLM. When encoding user embeddings, the recommender processes the user's historical interaction sequence ($N \geq 1$), and when encoding item embeddings, it processes only the item's ID ($N = 1$).

Given the collaborative embeddings $\mathbf{C} \in \mathbb{R}^{N \times d_c}$, M-Former selectively activates query experts via a router to extract collaborative information through multiple transformer layers. M-Former consists of K experts, each comprising L learnable vectors of size d_m, represented as $\mathbf{Q} \in \mathbb{R}^{L \times d_m}$. The router is a gating function with a learnable weight matrix $\mathbf{W}_r \in \mathbb{R}^{K \times d_c}$, responsible for computing the probability distribution of expert weights. Specifically, the weight for the j-th expert is computed as:

$$w_j = \frac{\sum_i \text{softmax}\left(\mathbf{C}\mathbf{W}_r^\top\right)_{ij}}{N}. \tag{4}$$

Following standard MoE architectures, only the top-k experts with the highest weights are activated. This enables M-Former to dynamically leverage different experts based on varying collaborative signals. The activated experts are then passed through Z layers of transformer blocks, where they extract the collaborative information contained in $\mathbf{C}$ using cross-attention:

$$\mathbf{Q}' = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{C}'^\top}{\sqrt{d_m}}\right)\mathbf{C}', \tag{5}$$

where $\mathbf{Q} \in \mathbb{R}^{L \times d_m}$ represents a selected query expert, and $\mathbf{C}' \in \mathbb{R}^{N \times d_m}$ is obtained by linearly mapping $\mathbf{C} \in \mathbb{R}^{N \times d_c}$. Finally, the experts are aggregated using their respective weights computed in Equation 4, and the aggregated result is then linearly mapped to the LLM's hidden size d to form the prefix, which is used to adapt the LLM with collaborative knowledge.

4.3 Model Learning

4.3.1 Loss Function

To ensure the balanced utilization of all experts in M-Former, we adopt a multi-task loss that incorporates both the recommendation and load-balancing objectives.

Specifically, the recommendation task is modeled using the item-item contrastive (IIC) loss (Li et al., 2023), which encourages the user and ground-truth item embeddings to be closer while pushing other irrelevant item embeddings further apart. The IIC loss is defined as:

$$\mathcal{L}_{\text{IIC}} = -\log \frac{e^{\cos(\mathbf{h}_u, \mathbf{h}_i^+)/\tau}}{\sum_{i \in \mathcal{I}} e^{\cos(\mathbf{h}_u, \mathbf{h}_i)/\tau}}, \tag{6}$$

where $\mathbf{h}_u$ and $\mathbf{h}_i$ represent the embeddings of user u and item i, $\mathbf{h}_i^+$ is the embedding of the ground-truth item, $\mathcal{I}$ represents the item set, and τ is a temperature hyperparameter.

To ensure efficient usage of all experts in M-Former, we also incorporate a load-balancing loss. This loss, following previous works (Lepikhin et al., 2021; Fedus et al., 2022), is computed as:

$$\mathcal{L}_{\text{LB}} = K \sum_{j=1}^{K} f_j w_j, \tag{7}$$

where K is the number of experts, w_j is the weight for the j-th expert calculated in Equation 4, and f_j is the fraction of items dispatched to the j-th expert by the router, calculated by:

$$p = \text{softmax}\left(\mathbf{C}\mathbf{W}_r^\top\right), \tag{8}$$

$$f_j = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{\text{argmax } p_i = j\}, \tag{9}$$

where $\mathbf{C} \in \mathbb{R}^{N \times d_c}$ represents the collaborative embeddings of the input item ID sequence of length N, $\mathbf{W}_r \in \mathbb{R}^{K \times d_c}$ is the weight matrix of the router, and $p \in \mathbb{R}^{N \times K}$ is the calculated score matrix, which represents the degree of correlation between the N items and the K query experts.

Finally, the overall loss function is a weighted sum of the item-item contrastive loss and the load-balancing loss:

$$\mathcal{L} = \mathcal{L}_{\text{IIC}} + \lambda \mathcal{L}_{\text{LB}}, \tag{10}$$

where λ is a hyper-parameter that balances the weight of different tasks.

4.3.2 Two-Phase Training

As shown in Equation 6, calculating the IIC loss requires the embeddings of all items in the item set $\mathcal{I}$, which are determined by the current model parameters at each training step. However, updating all item embeddings at each training step is computationally expensive and can lead to unstable supervision. Therefore, we introduce a two-phase training strategy. Specifically, in the first phase, item embeddings are updated only at the beginning of each epoch. The phase ends when optimal performance is achieved on the validation set. Then, the item embeddings from the epoch with the best validation performance are used throughout the second phase of training. During this phase, the model parameters are optimized to make the user embeddings closer to the fixed ground-truth item embeddings via the contrastive loss $\mathcal{L}_{\text{IIC}}$. Through this way, we reduce the computational cost and progressively optimize the item and user embeddings across two phases.

Datasets	#Users	#Items	#Inters.	Avg. n	Density
Scientific	11,041	5,327	76,896	6.96	1.3e-3
Arts	56,210	22,855	492,492	8.76	3.8e-4
Pet	47,569	37,970	420,662	8.84	2.3e-4
Games	11,036	15,402	100,255	9.08	5.9e-4

Table 1: Statistics of all datasets. Avg. n denotes the average number of items in the user interaction history.

5 Experiments

In this section, we conduct detailed experiments to demonstrate the effectiveness of our proposed Laser.

5.1 Experimental Settings

Datesets. We conduct experiments on four Amazon review datasets: "Industrial and Scientific", "Arts, Crafts and Sewing", "Pet Supplies", and "Video Games". Following previous works (Li et al., 2023; Hou et al., 2022), we use the five-core version and exclude items without titles. We collect user interactions and sort the items by timestamp. The statistics of the preprocessed datasets are shown in Table 1.

Baselines. We compare our Laser to a number of state-of-the-art baselines, including six traditional methods: SASRec (Kang and McAuley, 2018), BERT4Rec (Sun et al., 2019), RecGURU (Li et al., 2022), FDSA (Zhang et al., 2019), ZESRec (Ding et al., 2021), RECFORMER (Li et al., 2023), and three LLM-based methods: LLM4REC (Wang et al., 2024b), KAR (Xi et al., 2023), and LlamaRec (Yue et al., 2023). We list the details of these baselines in Appendix A.

Implementation Details. In this paper, we use BERT4Rec (Sun et al., 2019) to provide collaborative knowledge for LLMs and employ GPT2-Large (Radford et al., 2019), ChatGLM2-6B (GLM et al., 2024), and Llama2-7B (Touvron et al., 2023) as LLM backbones, which are also used in the three LLM-based baselines. The corresponding Laser

Models	Scientific			Pet			Arts			Games		
	R@10	N@10	MRR	R@10	N@10	MRR	R@10	N@10	MRR	R@10	N@10	MRR
Traditional Methods												
SASRec	13.11	8.03	7.12	8.81	5.69	5.07	13.42	8.48	7.42	9.53	5.47	5.05
BERT4Rec	10.61	7.90	7.59	7.65	6.02	5.85	12.36	9.42	8.99	10.48	6.54	6.07
RecGURU	7.81	5.75	5.66	4.15	3.66	3.71	7.42	5.25	4.88	4.79	3.86	3.96
ZESRec	12.60	8.43	7.45	10.24	7.64	7.25	13.49	9.70	8.70	8.44	5.30	5.05
RECFORMER	11.14	7.22	6.50	9.05	7.93	7.74	12.98	10.24	9.80	8.61	5.72	5.22
FDSA	9.67	7.16	6.92	9.49	6.73	6.50	12.09	9.94	9.41	9.31	6.00	5.46
LLM-based Methods												
Methods based on GPT2												
LLM4REC	12.57	7.64	6.83	9.18	7.69	6.81	12.66	9.27	8.80	8.57	5.46	5.13
Laser-G	12.91	8.42	7.68	9.83	7.97	7.45	13.42	10.31	9.72	9.12	5.83	5.34
Methods based on ChatGLM2												
KAR	12.65	8.94	8.13	9.42	7.24	6.77	13.57	9.17	8.18	9.44	5.82	5.26
Laser-C	**14.06***	9.83	9.16	11.21	**9.08***	8.61	**14.92***	**11.40***	**11.14***	10.75	7.13	6.46
Methods based on Llama2												
LlamaRec	12.75	8.57	7.93	9.61	7.54	7.11	13.68	8.60	7.94	9.58	5.79	5.41
Laser-L	13.87	**9.84***	**9.23***	**11.67***	9.05	**8.63***	14.53	10.96	10.81	**10.91***	**7.25***	**6.58***
Improv. (%)	**7.24**	**9.17**	**13.51**	**13.96**	**14.53**	**11.53**	**9.13**	**11.37**	**13.74**	**4.10**	**10.86**	**8.40**

Table 2: Performance comparison of different methods (all results are scaled by a factor of 100). The best results for Laser are marked in **bold**, and the best results for the baselines are underlined. Improv. indicates the improvements between Laser's and the baselines' best results, while * denotes statistically significant improvements (t-test with p-value < 0.05).

models based on these backbones are denoted as Laser-G, Laser-C, and Laser-L, respectively. Implementation details are provided in Appendix B.

Evaluation Settings. Following previous works (Li et al., 2023; Yue et al., 2023; Wang et al., 2024b), we evaluate using three common metrics: Recall@N, NDCG@N, and MRR. For data splitting, we apply the leave-one-out strategy (Kang and McAuley, 2018), where the most recent item in the interaction history is used for testing, the second for validation, and the rest for training. We report the average results on the test data.

5.2 Overall Performance

As shown in Table 2, Laser achieves substantial improvements across all metrics and datasets compared to all baselines. For instance, on the Pet and Arts datasets, Laser surpasses the best baseline in Recall@10 by 13.96% and 9.13%, respectively. Notably, Laser consistently outperforms LLM-based baselines using the same LLM backbones, demonstrating the effectiveness of our proposed Bi-Tuning framework with M-Former in seamlessly integrating collaborative information into LLMs and adapting them for controllable sequential recommendation.

Additionally, we note that Laser's performance improves as the LLM backbone scales. For example, on the Pet dataset, Laser-C (based on ChatGLM2-6B) and Laser-L (based on Llama2-7B) surpass Laser-G (based on GPT2-Large) in Recall@10 by 14.0% and 18.7%, respectively. This suggests that our model can be further improved with larger-scale LLM backbones.

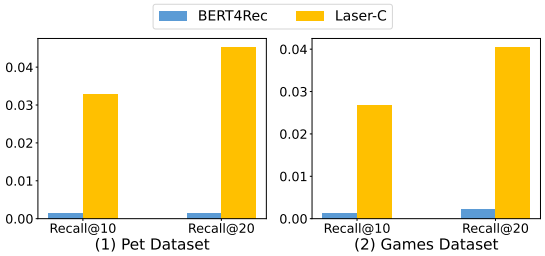

Figure 2: Performance comparison under the cold-start settings on the Pet and Games datasets.

5.3 Cold-Start Performance

To evaluate Laser's performance in cold-start scenarios, we compare Laser-C with BERT4Rec on the Pet and Games datasets. Following previous work (Kim et al., 2024), an item is categorized as "cold" if it falls within the bottom 35% of interactions. As shown in Figure 2, there is a clear performance gap between the two methods in both datasets, with Laser-C significantly outperforming BERT4Rec. This demonstrates that Laser effec-

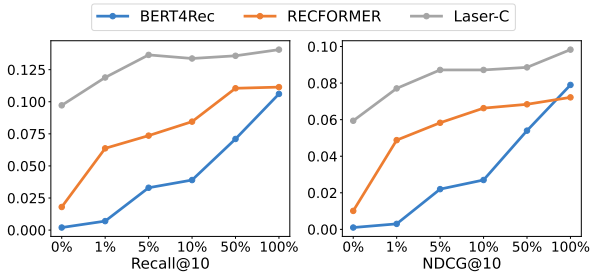

Figure 3: Performance comparison under the zero-shot and low-resource settings on the Scientific dataset.

	Recall@10	NDCG@10	MRR
Laser-C	**0.1406**	**0.0983**	**0.0916**
Bi-Tuning			
w/o prefix	0.1142	0.0716	0.0652
w/o suffix			
w/ average pooling	0.0579	0.0425	0.0481
w/ [EOS]	0.1004	0.0725	0.0683
M-Former			
w/o M-Former	0.1245	0.0844	0.0739
w/ one expert	0.1261	0.0889	0.0795
Model Learning			
w/o load-balancing loss	<u>0.1377</u>	<u>0.0893</u>	<u>0.0832</u>
w/o training phase 1	0.0793	0.0571	0.0571
w/o training phase 2	0.1320	0.0891	0.0774

Table 3: Ablation study on the Scientific dataset. The best results are in **bold** and the second are <u>underlined</u>.

tively leverages the language modeling and comprehension capability of LLMs to assist in recommending cold items.

5.4 Zero-Shot and Low-Resource Performance

To further demonstrate Laser's effectiveness, we conduct experiments in zero-shot and low-resource scenarios. Specifically, Laser-C, which combines semantic information with collaborative knowledge, is compared against two baselines: BERT4Rec (using only ID-based collaborative information) and RECFORMER (using only semantic information). These models, except for ID-based BERT4Rec, are trained on the Pet dataset and tested on the Scientific dataset with limited or no training data.

Figure 3 shows the results. We observe that: (1) Laser outperforms all baselines in the zero-shot scenario, achieving significantly better performance, despite seeing no data from the Scientific dataset. This performance is attributed to the Bi-Tuning framework, which fully leverages the generalization capabilities of LLMs for sequential recommendation. (2) Laser requires only 5% of the training data to outperform both baselines using 100% of the data. As training data increases, Laser's performance improves rapidly, demonstrating that minimal training data is sufficient for transferring Laser to an out-of-domain dataset, achieving better results than baselines requiring more data. This demonstrates the effectiveness of our framework in integrating collaborative knowledge into LLMs and adapting them for controllable and generalizable sequential recommender systems.

5.5 Ablation Study

To demonstrate the effectiveness of each module in Laser, we conduct comprehensive ablation studies on the Scientific dataset (Table 3) and Pet dataset (Appendix C). The results show that removing any module leads to a significant decrease in Laser's performance.

First, both the learnable prefix and suffix are crucial for adapting LLMs to sequential recommendation. Without the prefix, Recall@10, NDCG@10, and MRR drop significantly by 18.77%, 27.16%, and 28.82%, respectively, highlighting the prefix's role in adapting LLMs with collaborative information. Besides, we compare the suffix with two alternative strategies for generating item/user embeddings: (1) average pooling of all token embeddings from the LLM output and (2) replacing the learnable virtual suffix with a hard [EOS] token. Both variations result in noticeable performance degradation, with Recall@10, NDCG@10, and MRR decreasing by at least 28.6%, 26.2%, and 25.4%, respectively, demonstrating that the learnable suffix better converts the LLM output from language space to recommendation space to obtain high-quality item/user embeddings.

Second, the M-Former effectively integrates collaborative information into LLMs, significantly improving recommendation accuracy. Without the M-Former (where the prefix is randomly initialized), Recall@10, NDCG@10, and MRR drop by 11.45%, 14.14%, and 19.32%, respectively. Furthermore, with only one query expert, Recall@10, NDCG@10, and MRR decrease by 10.31%, 9.56%, and 13.21%, underscoring the importance of M-Former's adaptive expert utilization based on varying collaborative signals.

Finally, removing the load-balancing loss or either training phase reduces Laser's effectiveness, highlighting the importance of efficient expert utilization in M-Former and the necessity of two-

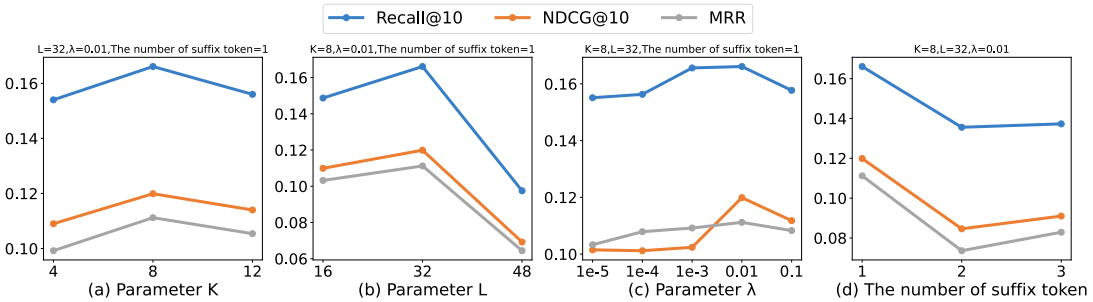

Figure 4: Comparison of different K, L, λ, and the number of suffix tokens on the validation set of the Scientific dataset based on Laser-C.

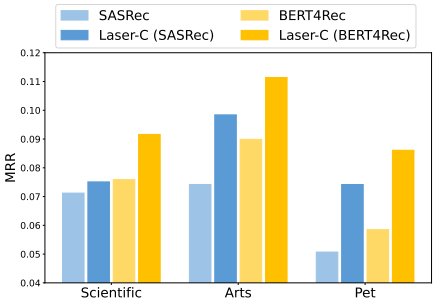

Figure 5: Performance comparison with different ID-based sequential recommender systems.

phase training, where the first phase learns high-quality item embeddings and the second optimizes user embeddings for accurate recommendations.

5.6 Further Discussion

We further discuss Laser focusing on the impact of different ID-based sequential recommenders and hyper-parameter settings. We also analyze the effects of various templates, with detailed results provided in Appendix D.

5.6.1 ID-based Sequential Recommender

We conduct additional experiments to study how the ID-based sequential recommender system influences Laser's performance. As shown in Figure 5, Laser-C's performance increases almost linearly with the performance of the employed ID-based recommender. For example, on the Pet dataset, Laser-C based on BERT4Rec surpasses Laser-C based on SASRec by 16.08%, while BERT4Rec outperforms SASRec by 15.38%. This indicates that Laser can be further improved by leveraging more powerful ID-based sequential recommender systems.

5.6.2 Parameter Analysis

We conduct a detailed parameter analysis, evaluating the number of query experts (K), the number of expert virtual tokens (L), the loss balance factor (λ), and the suffix token number. As shown in Figure 4, smaller values of K and L limit the M-Former's ability to extract collaborative information, while larger values increase the training complexity and reduce the model's effectiveness. Besides, Laser performs best with $\lambda = 0.1$, which balances the IIC and load-balancing losses. Additionally, a single suffix token performs best, as multiple tokens (whose output embeddings are averaged to form item/user embeddings) add complexity and reduce effectiveness. Finally, the optimal configuration is achieved with $K = 8$, $L = 32$, $\lambda = 0.1$, and a single suffix token.

6 Conclusion

In this paper, we propose Laser, a simple yet effective Bi-Tuning framework with collaborative information for controllable large language model-based sequential recommendation. Paritcularly, we introduce Bi-Tuning, an efficient fine-tuning method that adapts LLMs to sequential recommendation via a learnable prefix and suffix. The prefix effectively incorporates collaborative information, while the suffix transforms LLM output into item/user embeddings for similarity comparison, enabling controllable recommendations. To better integrate ID-based collaborative information, we introduce a lightweight MoE-based querying transformer that activates different experts to extract relevant information from varying collaborative signals of frozen ID-based recommenders, paired with a multi-task loss for load-balancing. Finally, a two-phase training strategy is used to progressively obtain high-quality item and user embeddings through the learnable suffix. Extensive experiments on real-world datasets show that Laser effectively adapts LLMs to sequential recommendation tasks, substantially outperforming state-of-the-art methods.

7 Limitations

In this work, we validate the effectiveness of Laser on current mainstream decoder-only LLMs, such as GPT2-Large, ChatGLM2-6B, and Llama2-7B. However, a limitation remains in that we do not explore how Laser can be adapted to other LLM architectures, such as encoder-decoder models, which may offer different benefits for sequential recommendation tasks. In future work, we plan to investigate how Laser can be integrated with these alternative architectures to further enhance its performance and applicability.

References

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*, pages 1007–1014.

Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, and et al. 2021. Sequential recommendation with graph neural networks. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 378–387.

Zheng Chen, Ziyan Jiang, Fan Yang, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Aram Galstyan. 2023. Graph meets LLM: A novel approach to collaborative filtering for robust conversational understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP 2023 - Industry Track, Singapore, December 6-10, 2023*, pages 811–819. Association for Computational Linguistics.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling.

Hao Ding, Yifei Ma, Anoop Deoras, Yuyang Wang, and Hao Wang. 2021. Zero-shot recommender systems.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, pages 120:1–120:39.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, and et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Ruining He and Julian J. McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *IEEE 16th International Conference on Data Mining (ICDM) (2016)*.

Balázs Hidasi, Alex, ros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *4th International Conference on Learning Representations*.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian J. McAuley, and et al. 2024. Large language models are zero-shot rankers for recommender systems. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval*, pages 364–381.

Wang-Cheng Kang and Julian J. McAuley. 2018. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining*, pages 197–206.

Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Min-Chul Yang, and Chanyoung Park. 2024. Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 1395–1406. ACM.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, and et al. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Chenglin Li, Mingjun Zhao, Huanming Zhang, Chenyun Yu, Lei Cheng, Guoqiang Shu, and et al. 2022. Recguru: Adversarial learning of generalized user representations for cross-domain recommendation. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining*, pages 571–581.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and et al. 2023. 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.

Jiacheng Li, Yujie Wang, and Julian J. McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining*, pages 322–330.

Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1419–1428.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4582–4597.

Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, and et al. 2024. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3497–3508.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and et al. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Wensheng Lu, Jianxun Lian, Wei Zhang, Guanghua Li, Mingyang Zhou, Hao Liao, and Xing Xie. 2024. Aligning large language models for controllable recommendations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 8159–8172. Association for Computational Linguistics.

Junyan Qiu, Haitao Wang, Zhaolin Hong, Yiping Yang, Qiang Liu, and Xingxing Wang. 2023. Controlrec: Bridging the semantic gap between language model and personalized recommendation. *CoRR*, abs/2311.16441.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and et al. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450.

Jiaxi 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*, pages 565–573.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, and et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Hanbing Wang, Xiaorui Liu, Wenqi Fan, Xiangyu Zhao, Venkataramana Kini, Devendra Yadav, Fei Wang, Zhen Wen, Jiliang Tang, and Hui Liu. 2024a. Rethinking large language model architectures for sequential recommendations. *CoRR*, abs/2402.09543.

Xinyuan Wang, Liang Wu, Liangjie Hong, Hao Liu, and Yanjie Fu. 2024b. Llm-enhanced user-item interactions: Leveraging edge information for optimized recommendations.

Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xianling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 169–178.

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 346–353.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, and et al. 2023. Towards open-world recommendation with knowledge augmentation from large language models.

Zhengyi Yang, Jiancan Wu, Yanchen Luo, Jizhi Zhang, Yancheng Yuan, An Zhang, and et al. 2023. Large language model can interpret latent space of sequential recommender.

Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. 2023. Llamarec: Two-stage recommendation using large language models for ranking.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023a. Recommendation as instruction following: A large language model empowered recommendation approach.

Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, and et al. 2019. Feature-level deeper self-attention network for sequential recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 4320–4326.

Yang Zhang, Keqin Bao, Ming Yan, Wenjie Wang, Fuli Feng, and Xiangnan He. 2024. Text-like encoding of collaborative information in large language models for recommendation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9181–9191. Association for Computational Linguistics.

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

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*, pages 1435–1448. IEEE.

A Baselines

To comprehensively evaluate the performance of our proposed Laser, we compare it to state-of-the-art baselines, including six traditional methods and three LLM-based methods.

(1) Traditional Baselines:

- **SASRec** (Kang and McAuley, 2018) employs a self-attention mechanism to capture the semantic relevance between the user interaction sequence and the candidate items.
- **BERT4Rec** (Sun et al., 2019) is a bidirectional self-attentive model, employing the cloze objective to model users' dynamic preferences from their historical behaviors.
- **RecGURU** (Li et al., 2022) introduces an adversarial learning method to incorporate user information across domains and obtain generalized user representations for sequential recommendation.
- **FDSA** (Zhang et al., 2019) proposes a feature-level self-attention network that integrates different heterogeneous features of items into feature sequences with different weights through a vanilla attention mechanism.
- **ZESRec** (Ding et al., 2021) utilizes a pre-trained language model to convert item descriptions into feature representations.
- **RECFORMER** (Li et al., 2023) formulates items as key-value attribute pairs and utilizes pre-trained language models to encode them for ID-free sequential recommendation.

(2) LLM-based Baselines:

- **LLM4REC** (Wang et al., 2024b) incorporates collaborative information into the LLM's attention weight calculation. Besides, it adds user and item IDs to the LLM's vocabulary for recommendation and defines a series of pre-training and supervised fine-tuning tasks to help the LLM learn the meanings of these IDs in the context of recommendations. GPT2 (Radford et al., 2019) is used as the backbone.
- **KAR** (Xi et al., 2023) proposes a hybrid-expert adapter that condenses LLM-generated world knowledge to enhance the performance of recommendation models. Following the

original paper, we use ChatGLM2-6B (GLM et al., 2024) to provide the dense vector outputs containing world knowledge.

- **LlamaRec** (Yue et al., 2023) leverages a two-stage framework, where a traditional sequential recommender model is first used for retrieval, followed by ranking a small set of candidate items using Llama2-7B (Touvron et al., 2023).

B Implementation Details

In this work, we use a pre-trained BERT4Rec (Sun et al., 2019) to provide collaborative knowledge for the LLM, with the number of transformer blocks, attention heads, and the dimension of each attention headset to 2, 2, and 32, respectively. We employ GPT2-Large (Radford et al., 2019), ChatGLM2-6B (GLM et al., 2024), and Llama2-7B (Touvron et al., 2023) as LLM backbones, which are also used in the three LLM-based baselines. The M-Former consists of 12 transformer blocks, with alternate blocks performing cross-attention. The hidden size and number of attention heads are set to 768 and 12, respectively. The expert number K is set to 8, with only the top 3 experts activated, and the query token number L is set to 32. Additionally, the router is implemented as a fully connected layer, with input and output dimensions set to 64 and 8, respectively.

During training, the BERT4Rec and LLM backbones are frozen, while the other learnable modules are randomly initialized and trained in two phases (as described in Section 4.3.2). We set the batch size to 4, the learning rate to 1e-4, the loss weight hyperparameter λ to 0.01, and the temperature hyperparameter τ to 0.05. The Adam optimizer is used, and the Laser is trained for 15 epochs in the first phase and 5 epochs in the second phase.

C Ablation Study on the Pet Dataset

As shown in Table 4, the ablation results on the Pet dataset indicate that removing any module significantly decreases Laser's performance, which is consistent with the results on the Scientific dataset.

D Influence of the Template

In this work, we utilize a unified template to organize both the user interaction history and the single item information, which is shown in Section 4.1. The template instructs LLMs to summarize the semantic information into the suffix, which is further used for recommendation. To ensure the template's plausibility, we compare it with three other variants, including: (1) deleting the specified phrase "into a single token", (2) deleting the entire instruction "You are an intelligent ... the user has browsed the following items:", (3) using a different instruction for item embedding generation, "You are an intelligent recommendation assistant. Please summarize the item characteristics into a single token:". As shown in Table 5, compared to the other three variants, our prompt template can significantly improve Recall@10, NDCG@10, and MRR by more than 30.06%, 18.43%, and 15.22%, respectively. This demonstrates the effectiveness of our prompt template in harnessing the powerful capabilities of LLMs with clear, consistent, and appropriate instruction.

	Recall@10	NDCG@10	MRR
Laser-C	**0.1406**	**0.0983**	**0.0916**
Bi-Tuning			
w/o prefix	0.0875	0.0701	0.0653
w/o suffix			
w/ average pooling	0.0470	0.0453	0.0477
w/ [EOS]	0.0758	0.0637	0.0645
M-Former			
w/o M-Former	0.1049	0.0775	0.0721
w/ one expert	0.1056	0.0818	0.0763
Model Learning			
w/o load-balancing loss	0.1088	<u>0.0887</u>	0.0809
w/o training phase 1	0.0531	0.0492	0.0418
w/o training phase 2	<u>0.1091</u>	0.0853	<u>0.0812</u>

Table 4: Ablation study on Pet dataset. The best results are in bold and the second best results are underlined.

Templates	Recall@10	NDCG@10	MRR
Laser-C	**0.1406**	**0.0983**	**0.0916**
w/o specified phrase	<u>0.1081</u>	<u>0.0830</u>	<u>0.0795</u>
w/o instruction	0.0993	0.0767	0.0654
w/ two instructions	0.0968	0.0640	0.0571

Table 5: Performance comparison under different hard prompt templates on the Scientific dataset.