Sequential LLM Framework for Fashion Recommendation

Han Liu[♠†], **Xianfeng Tang**[♢], **Tianlang Chen**[♢], **Jiapeng Liu**[♢],
Indu Indu[♢], **Henry Peng Zou**[♣†], **Peng Dai**[♢], **Roberto Fernandez Galan**[♢],
Michael D Porter[♢], **Dongmei Jia**[♢], **Ning Zhang**[♠], **Lian Xiong**[♢]
[♢]Amazon [♠]Washington University in St. Louis [♣]University of Illinois Chicago
{h.liu1,zhang.ning}@wustl.edu, {xianft,ctianlan,liujiape,indchand}@amazon.com,
pzou3@uic.edu, {pengdai,galanrob,mdporter,djia,lianxion}@amazon.com

Abstract

The fashion industry is one of the leading domains in the global e-commerce sector, prompting major online retailers to employ recommendation systems for product suggestions and customer convenience. While recommendation systems have been widely studied, most are designed for general e-commerce problems and struggle with the unique challenges of the fashion domain. To address these issues, we propose a sequential fashion recommendation framework that leverages a pre-trained large language model (LLM) enhanced with recommendation-specific prompts. Our framework employs parameter-efficient fine-tuning with extensive fashion data and introduces a novel mix-up-based retrieval technique for translating text into relevant product suggestions. Extensive experiments show our proposed framework significantly enhances fashion recommendation performance.

1 Introduction

In recent years, fashion e-commerce has garnered considerable global attentions from both consumers and investors. By 2023, the U.S. retail fashion e-commerce market is projected to generate revenues exceeding 207 billion U.S. dollars (Gelder). One of the primary objectives of e-commerce is to provide a smooth purchase experience for consumers to purchase products they are looking for. To this end, recommendation systems (RS) have become an essential part of many businesses (Ying et al., 2018; Zhang et al., 2019). While existing fashion recommendation systems (He and McAuley, 2016b; Liu et al., 2017; Kang et al., 2017; Yu et al., 2021) predominantly incorporate the visual appearance into the traditional recommendation, they often struggle to capture the evolving nature of user interactions over time. In light of this, there has been a growing interest in sequential recommendation techniques (Sun et al., 2019; Kang and McAuley, 2018; Li et al., 2023). These techniques model historical user interactions as temporally ordered sequences, thereby achieving remarkable efficacy in capturing both short-term and long-term user preferences.

While sequential recommendations have succeeded in general e-commerce, the fashion domain poses unique challenges. Our analysis of real-world user interactions on Amazon fashion highlights key differences: First, the rapid fashion turnover leads to a sparse user-item interaction matrix, intensifying the cold-start problem (Liu et al., 2020). Second, extensive purchase comparisons demand sophisticated approaches to capture fine-grained user preferences. Third, fashion-specific attributes like seasonality, occasion, and holiday trends require specialized modeling. Fourth, diverse search queries that reflect explicit user intentions, necessitate novel modeling techniques. Beyond these fashion-specific challenges, traditional recommendation contexts often require specialized models tailored to particular scenarios, such as the cold-start problem (Dong et al., 2020), which will result in a large number of models that are challenging to maintain and scale.

To tackle these challenges holistically, we present a sequential fashion recommendation system augmented by a large language model (LLM). Trained on vast and diverse datasets, LLMs have a profound understanding of various domains. Leveraging their extensive knowledge and commonsense reasoning capabilities (Zhao et al., 2023), LLMs provide a promising solution to generate meaningful recommendations. This is particularly beneficial in overcoming cold start problems and in accurately discerning fine-grained user preferences. Additionally, LLMs could offer a unified framework capable of addressing diverse recommendation tasks. Our LLM-augmented recommendation system consists of three primary stages. In the first stage, prompt engineering techniques are

[†]Work done as an intern at Amazon.

used to devise specialized prompts that align with recommendation-specific goals, enabling LLM to perceive fine-grained user preferences. In the second stage, we adapt Parameter-Efficient Fine-Tuning (PEFT) techniques (Hu et al., 2021; Dettmers et al., 2023) to mitigate prohibitively expensive training costs. In the final stage, we utilize predicted product titles and IDs to retrieve and rank potential candidate items. We present a mix-up-based retrieval technique that harnesses the strengths of both ID and title embeddings. Our contributions can be summarized as follows:

- We conduct an in-depth data analysis on real-world user interaction patterns, identifying four key characteristics for fashion recommendation.
- We propose a comprehensive recommendation framework tailored to the fashion domain. Within this framework, we propose advanced LLM enhancement techniques to address the unique challenges for fashion recommendation.
- The comprehensive evaluations demonstrate that the proposed framework significantly enhance recommendation performance.

2 Related Work

Sequential Recommendation. Recommendation systems have gained significant interest from both academia and industry (Ma et al., 2022), with sequential recommendation receiving particular attention due to its exceptional capabilities of capturing the long-term and short-term dynamics of users (Li et al., 2022; Ma et al., 2023). The objective of sequential recommendation is to predict the next items that users may be interested in based on their historical interactions. There are various techniques being proposed to model user sequential patterns, from the Markov Chain (He and McAuley, 2016a; Rendle et al., 2010) in early works to recent neural network-based techniques, such as Gated Recurrent Units (GRU) (Hidasi et al., 2015), Convolutional Neural Network (CNN) (Tang and Wang, 2018), and Transformer (Sun et al., 2019; Kang and McAuley, 2018; Hou et al., 2022). Recently, Recformer (Li et al., 2023), a Transformer-based framework for learning transferable language representations, has been proposed for sequential recommendations. It has shown superior performance, especially in cold-start settings.

Fashion Recommendation. Fashion recommendation systems, which target one vertical market - fashion and garment products, have gained popularity recently (Lin et al., 2019; Hou et al., 2019). Existing approaches primarily utilize visual signals to capture fashion characteristics. Specifically, visual features have been integrated into fashion recommendations to enhance item representations (He and McAuley, 2016b; He et al., 2016; Kang et al., 2017), model visual compatibility relationships (Chen et al., 2019; Yin et al., 2019), and identify aesthetic and style information (Yu et al., 2021). For example, He and McAuley (2016b) propose the Visual Bayesian Personalized Ranking (VBPR), which incorporates visual features extracted from product images into matrix factorization frameworks using pre-trained CNNs. Yin et al. (2019) utilized visual encodings to learn visual compatibility by training a triplet network, where an anchor item is paired with both a compatible and a non-compatible item to learn embeddings that capture visual compatibility. Additionally, Yu et al. (2021) introduced a deep aesthetic network that extracts aesthetic features from product images, incorporating them into recommendations to model users' preferences for aesthetic appeal. The methods for extracting visual signals have evolved over time. Early studies typically used pre-trained CNNs for visual encodings (He et al., 2016; He and McAuley, 2016b). However, recent works have shifted towards training visual encoders on specialized datasets (Yin et al., 2019) or jointly training visual feature extractors and recommendation modules (Kang et al., 2017; Lin et al., 2019).

While incorporating visual signals is an inspiring direction, it falls outside the scope of and is furthermore orthogonal to our current study, which focuses on leveraging textual data to model user interactions. This choice is driven by the fact that learning effective product representations from images typically requires large datasets to generalize well (Deldjoo et al., 2022), which would introduce significant demands in terms of data collection and computational resources, making it challenging for industrial deployment.

3 Fashion Characteristics

Fashion-related shopping presents unique characteristics that must be carefully considered when developing RS. We conduct an in-depth analysis on real-user interaction patterns in Amazon Fashion, and identify the following key characteristics:

C1: High Turnover of Products. The fashion domain is characterized by a rapid turnover of items,

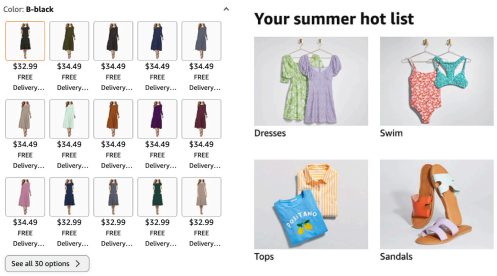

Figure 1: Examples highlighting fashion characteristics. Figure (a) illustrates the extensive color variations in fashion products, while Figure (b) demonstrates the seasonality attributes of fashion items.

introducing a continuous stream of unique new products to platforms. For instance, Amazon Fashion adds approximately 3 million new purchasable products each month. Additionally, the volume of new fashion items significantly exceeds that of other categories, being 3.6 times greater than new electronics and 6.7 times more than new toy products on Amazon. This constant influx leads to a notably sparse user-item interaction matrix, gives rise to the cold-start problem (Liu et al., 2020).

C2: Thorough Purchase Comparisons. Users involved in fashion-related purchases tend to engage in more comprehensive comparisons than those shopping in other categories. For example, the average interaction length for fashion-related purchases is 55% longer than for electronics and 81% longer than for toy-related purchases. These comprehensive comparisons can be attributed to the extensive range of options—colors, styles, and sizes of fashion products. Figure 1 (a) provides an example of a typical shopping page for women's dresses, which offers 30 different color options.

C3: Fashion Attribute-Driven Shopping. Fashion items often come with distinct attributes such as seasonality, occasion, and holiday-specific trends, which significantly influence user shopping intention. For instance, Figure 1 (b) shows a selection of items popular in summer, which might not receive the same attention in winter.

C4: High Diversity of Search Queries. Search queries serve as a crucial context for understanding the evolving interests of users. We analyze the average volume of unique search queries over multiple days across three months. Our analysis shows that the number of unique search queries for fashion items is, on average, 2.63 times as much as electronics and 2.38 times as much as toys.

We highlight that while other industries may share some characteristics we've identified, the simultaneous presence of all four is unique to the fashion industry. Additionally, the importance of each characteristic in fashion differs from other domains. For example, attributes like seasonality, occasion, and trends have a more fine-grained influence on user choice in fashion compared to electronics or consumable products. In fashion, these factors influence not only availability but also social desirability and attractiveness at a given time.

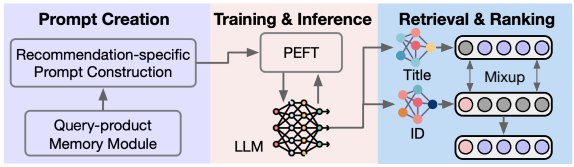

Figure 2: An overview of our method.

4 Method

4.1 Problem Formulation and Overview

Problem Formulation. In the realm of sequential recommendation, consider a system composed of a set of users and items. The set of users is represented by $\mathcal{U} = \{u_1, u_2, ..., u_N\}$, the set of items by $\mathcal{V} = \{v_1, v_2, ..., v_M\}$ and the set of queries as $\mathcal{Q} = \{q_1, q_2, ..., q_S\}$. Each user $u_i \in \mathcal{U}$ is associated with an interaction sequence S_i, which can be denoted as $S_i = [(s_{1,i}, a_{1,i}, t_{1,i}), ..., (s_{K,i}, a_{K,i}, t_{K,i})]$, where K is the sequence length, $a_{k,i}$ represents the specific action type, $t_{k,i}$ represents the timestamp of the action, $s_{k,i}$ can be an item or a query depending on the action type. When $a_{k,i}$ represents a search action, $s_{k,i} \in \mathcal{Q}$ represents the k-th query. For other actions, $s_{k,i} \in \mathcal{V}$ represents the k-th interacted item. In this paper, we are interested in three action types, search, click, and purchase behavior, and we aim to predict the future item the user will be interested in purchasing after observing interaction sequences S_i. Additionally, each item v is associated with an attribute dictionary containing various textual information, such as titles, colors, and descriptions. We formulate these as key-value attribute pairs and assign a unique ID to each item, in line with ID-based recommendation methods (Sun et al., 2019; Kang and McAuley, 2018).

Challenges and Overview. Addressing the unique characteristics of fashion poses significant challenges for recommendation systems. For instance, the cold-start problem remains a persistent issue in recommender systems. Traditional approaches to mitigate this challenge often rely on complex and

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
Instruction: Given the sequence of user interactions and product attributes, recommend a product that the user is most likely to purchase. Take into account the **sequential order** of the actions, as they reflect the user's shopping mission. Focus particularly on **product attributes** such as color and size, **season**, **occasion**, and **holiday**, as these significantly influence user decisions. **Prioritize recommendations from the candidate product ids provided in the input.** Directly predict the **product id** and **product title** as output. The output must strictly follow {<id>id</id>,<title>title</title>} format."
###Input: User search **keyword** <sleeveless blouses for women dressy>, then click product {"**id**": "41", "**title**": "Timeson Black Tunic Tops For Women,Chiffon Blouses For Women Business Casual Summer Tanks Shirts For Office Work Black M", "**category**": "Blouses", "**brand**": "Timeson", "**color**": "Black", "**size**": "M", "**season**":"Summer", "**holiday**":"", "**occasion**": "Business, Causal, Office,Work"}. Candidate product ids:[73818,6878,3,3096,1864,45884].
###Response: {<id>258</id>,<title>RUBZOOF Women's Sleeveless Chiffon Tank Tops Casual Summer V Neck Trendy Blouses Black M</title>}

Figure 3: The demonstration example of our prompt.

specialized architectures (Zhu et al., 2021; Dong et al., 2020). Capturing fine-grained user preferences further complicates this task, typically requiring specialized modules (Wang et al., 2019; Chen et al., 2021). Additionally, leveraging search data to enhance recommendations remains relatively unexplored (Si et al., 2023), where the primary difficulty lies in the distinct nature of user intent in search versus recommendation tasks. To address these challenges, we propose an LLM-augmented fashion sequential recommendation system, as shown in Figure 2. The process initiates with the creation of prompts. A query-product memory module assesses user-item interactions to identify top products associated with user queries. This information is synthesized into a natural language format using a recommendation-specific prompt template, incorporating fashion-related attributes. In the subsequent training stage, we utilize the prepared prompts to fine-tune the LLM through a Parameter-Efficient Fine-Tuning method. Finally, in the retrieval and ranking stage, we convert the generated titles and IDs into embeddings using two specialized models. These embeddings are integrated into a retrieval module with a mixup strategy, obtaining the final recommended items.

4.2 Prompt Design

Recommendation-specific Prompt Construction. Prompting offers a natural and intuitive interface for humans to interact with LLMs (Zhou et al., 2022). Given that LLMs are initially trained for general tasks, specialized prompts are essential for aligning LLMs with recommendation-specific goals. A demonstration example of our designed prompt is given in Figure 3.

The instruction segment aims to clearly define the task and consists of three core elements: *task description* (highlighted in purple), *execution requirement* (highlighted in blue), and *format indicator* (highlighted in brown). In the task description, we explicitly specify that the context is a recommendation task. The execution requirement emphasizes a set of strategies tailored to address the unique characteristics of fashion. A prime emphasis here is the consideration of sequential order. To address *C2*, we intend for the LLM to focus on varying attributes, as they offer insight into users' fine-grained preferences. To address *C3*, we emphasize the importance of fashion-specific attributes. To address *C4*, we observed that customers generally have preferences in purchasing the top exposure results on the shopping page, thus we instruct the LLM to prioritize the recommendation in the top exposure results corresponding to the search query. Finally, the format indicator strictly defines an output format for automated decoding. The input segment is a refined representation of user-item interactions, enriched with detailed item attributes. Search queries are also included to highlight their importance in the recommendation task. The response segment, employed only during the training phase, encapsulates the final item purchased by the user, including both the product ID and title.

Query-product Memory Module. We observe that users frequently opt to purchase items listed at the top of their search results. In response to this behavior, we propose a Query-Product Memory Module that preserves key-value pairs consisting of search queries and their corresponding product listings. To obtain these product lists, data is grouped by search queries and then sorted by organic position. Recognizing that queries can appear in various forms that convey similar meanings, we employ CLIP (Radford et al., 2021) to convert these queries into embedding vectors, which serve as the keys in our module. During the recommendation process, the current search query is transformed into its respective embedding, enabling us to compute the cosine distance, identify the nearest Q matching queries, and subsequently retrieve their associated top V products.

4.3 Training Strategy

Low-Rank Adaptation (LoRA) (Hu et al., 2021) has emerged as a notable Parameter-Efficient Fine-Tuning (PEFT) technique, offering performance comparable to full fine-tuning while requiring substantially fewer trainable parameters. Consequently, we have adopted this method to fine-tune

our model. We further reduce memory usage by employing model quantization as implemented in QLoRA (Dettmers et al., 2023). Specifically, we maintain distinct storage and computation data types. We quantize the model to a more memory-efficient storage type and, during the forward and backward passes, dequantize the data back to the computation type to avoid performance loss. During our preliminary experiments, we observed that the model exhibited a 7% likelihood of generating outputs in an inconsistent format. This inconsistency made the automated decoding of product IDs and titles challenging. One possible reason is that product titles in e-commerce often display limited sentence coherence and are more like a collection of individual words, setting them apart from typical natural language structures. To mitigate this issue, we identified the high-perplexity prompts and subjected them to additional training cycles relative to their lower-perplexity counterparts.

4.4 Retrieval and Ranking Method

Title Embedding Model. To effectively capture the semantic similarity of item titles in recommendation tasks, we leveraged the insight that the items that were purchased in the same search queries should be similar in embedding space. Based on this insight, we first tokenize both the query and product title. Once tokenized, the model computes the embeddings for query and title by employing the LSTM model. We train the whole model using a triplet loss (Jiang et al., 2016), where we pair two hard negatives with one positive pair during each forward pass. The positive match means the item that was purchased from the query. We choose the title that is closest to the query but is not a positive match and the query that is closest to the title but is not a positive match as the hard negatives.

ID Embedding Model. The ID embedding model maps pre-defined item IDs into their embeddings. We leverage the item embedding table from the CORE model, which has demonstrated superior performance compared to state-of-the-art methods. Specifically, we train the CORE model using user-item interaction sequences, then keep only the item embedding table as our ID embedding model.

Retrieval with Mixup. After obtaining both title embeddings and ID embeddings, the next step is to perform retrieval and ranking processes to get the candidate items for recommendation. Title embeddings are designed to capture the semantic content of an item's title, thus offering better generalization, even if an item hasn't been seen before (*i.e.* cold start). Conversely, ID embeddings are designed to uniquely represent specific items, so the embedding can capture nuances specific to that item, thus being suitable in top matches. To effectively combine the advantages of the two methods, we propose a mixup-based retrieval method. This approach begins with separate retrievals based on title and ID embeddings, resulting in two distinct lists of items. To generate our final list of top-K items, we adopt the following approach: We select the top-N items from the ID embedding-based list. Subsequently, we choose items ranging from positions $N+1$ to K from the title embedding-based list. We set $N = 1$ for all our experiments.

5 Experiments

5.1 Experimental Setup

Datasets. We have collected a large-scale dataset derived from customer interactions on the Amazon fashion service, containing approximately 5.9 million user shopping interactions with a total of 2.4 million products. We aggregated them into four primary categories: Luggage and Bags, Footwear, Accessories and Jewelry, and Clothing. The sequences included three action types: search, click, and purchases. We also filtered the 'click' interactions on the items that were eventually purchased. Each item in our dataset is described by an array of attributes such as item and user identifier, product title, category, brand, color, and size. The statistics of the data after processing are given in Table 1.

Table 1: Statistics of the datasets. Avg. Len. represents the average length of interaction sequences.

Dataset	#Users	#Items	#Inters.	Avg. Len.	Density
Lug. & Bags	10,611	61,550	131,647	12.41	2.02E-04
Footwear	63,273	380,385	714,628	11.29	2.97E-05
Acc. & Jew.	106,104	524,433	1,376,999	12.98	2.47E-05
Clothing	274,285	1,386,910	3,635,414	13.25	9.56E-06

Evaluation Settings. To assess the efficacy of our sequential recommendation approach, we employ three widely used metrics: Recall@N, NDCG@N, and MRR, where N is set to 10. During the evaluation, we rank the ground-truth item (*i.e.*, final purchased item) of each sequence among all items in the same category and report the average values of all sequences in the test data. We employ the common leave-one-out strategy (Sun et al., 2019; Zhou et al., 2020) to split the data for evaluation.

Table 2: Performance comparison of our method with the state-of-the-art methods across different datasets.

Method	Luggage & Bags			Footwear			Accessories and Jewelry			Clothing		
	Recall@10	NDCG@10	MRR	Recall@10	NDCG@10	MRR	Recall@10	NDCG@10	MRR	Recall@10	NDCG@10	MRR
GRU4Rec	0.0336	0.0221	0.0185	0.0185	0.0124	0.0105	0.0230	0.0155	0.0131	0.0221	0.0155	0.0134
SASRec	0.1015	0.0613	0.0487	0.0857	0.0548	0.0452	0.1440	0.0825	0.0631	0.1485	0.0827	0.0617
BERT4Rec	0.0975	0.0600	0.0485	0.1405	0.0770	0.0568	0.0904	0.0580	0.0479	0.0676	0.0435	0.0361
NextItNet	0.0176	0.0094	0.0069	0.0123	0.0103	0.0097	0.0185	0.0150	0.0139	0.0111	0.0087	0.0079
CORE	0.2612	0.1404	0.1027	0.3075	0.1566	0.1092	0.2493	0.1327	0.0962	0.1989	0.1008	0.0699
Recformer	0.2577	0.1692	0.1455	0.2181	0.1352	0.1161	0.1719	0.1132	0.1004	0.1741	0.1102	0.0972
Recformer w/ query	0.2642	0.1834	0.1524	0.2325	0.1436	0.1225	0.1990	0.1220	0.1046	0.1888	0.1276	0.1059
Ours	**0.2786**	**0.2037**	**0.1804**	**0.3377**	**0.1791**	**0.1470**	**0.2624**	**0.1676**	**0.1343**	**0.2593**	**0.1658**	**0.1440**

Baselines. To evaluate the performance of the proposed method, we compare it with the following representative baselines: GRU4Rec (Hidasi et al., 2015), SASRec (Kang and McAuley, 2018), BERT4Rec (Sun et al., 2019), NextItNet (Yuan et al.), CORE (Hou et al., 2022), and Recformer (Li et al., 2023). To ensure a more fair comparison, we also compare with Recformer w/ query, which is similar to Recformer, with the only change of adding the search query as part of the input.

Implementation Details. We select Falcon-7b (fal) as our base LLM model. We implemented the training framework by using Huggingface PEFT library[1]. For LoRA, we set rank r to 16, scaling parameter α to 16, and dropout rate to 0.05. The maximum number of tokens for each interaction sequence is 1024. The models were trained on 8 Nvidia Tesla V100 GPUs. We optimized Falcon with AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate 2e-5. We only fine-tuned the model for 1 epoch, except in cases involving prompts with high perplexity, where we selected the top 20% of prompts as high perplexity prompts for fine-tuning 3 epochs. During the generation, we set the max new tokens to 64, the temperature to 0.05, and the probability threshold of nucleus sampling to 0.95. For the implementation of Recformer, we followed the official Github repository[2]. For all other baselines, we followed the suggested settings and implementations in RecBole (Zhao et al., 2021). To ensure a fair comparison, we conducted extensive hyperparameter tuning for each baseline method across different datasets.

5.2 Evaluation Results

We compared the performance of our method to baselines on four different datasets, the results are given in Table 2. Our method achieves the best overall performance on all datasets. Notably, we observed a 9.8% improvement in Recall@10 and a 14.4% improvement in NDCG@10 on the footwear dataset. On sparser datasets, the gains are more significant, with our method achieving a 30.4% improvement in Recall@10 and a 64.5% improvement in NDCG@10 on the clothing dataset. This is because item IDs cannot capture the rich semantic relationships that are readily expressed in item texts (*e.g.*, color, brand). In comparison to Recformer, which also leveraged text information, our method has the additional advantage of incorporating general knowledge and reasoning capabilities inherent in large language models. This yielded superior performance across recommendation tasks.

Table 3: Ablation study of different design components.

Variants	Recall@10	NDCG@10	MRR
Ours	0.2786	0.2037	0.1804
w/o product attributes	0.2293	0.1396	0.1117
w/o query-product memory	0.2595	0.1786	0.1518
w/o text embedding	0.1757	0.1569	0.1510
w/o id embedding	0.2412	0.1480	0.1189
w/ CLIP text embedding	0.2004	0.1245	0.1039
w/o q.p.m. & id emb.	0.2343	0.1424	0.1139
w/o q.p.m. & id. & pro. a.	0.2184	0.1285	0.1006

5.3 Ablation Study

We analyzed how different components in our design influence recommendation performance by introducing various model variants and testing them on the luggage and bags dataset. Specifically, we consider the following variants: (1) w/o product attributes: Product representation includes only the product title, omitting all attributes. (2) w/o query-product memory: Removes the query-product memory module. (3) w/o text embedding: Uses only ID embeddings for item retrieval. (4) w/o ID embedding: Uses only text embeddings for

[1]PEFT: https://huggingface.co/docs/peft/index
[2]Recformer: https://github.com/JiachengLi1995/Recformer

item retrieval. (5) w/ CLIP text embedding: Uses CLIP models for item retrieval. (6) w/o q.p.m. & id emb.: a combination of the removals from variants (2) and (4). (7) w/o q.p.m. & id. & pro. a.: combining the removals from variants (1), (2), and (4). The results in Table 3 show that each component improves performance. Notably, variants 3 and 4 highlight the benefits of our mixup-based retrieval method. The performance gap between variants 4 and 5 indicates that CLIP embedding models are less effective for recommendation tasks. Additionally, the slight performance drop from (4) to (6) indicates that the Query-Product Memory Module mainly influences the ID representation. Comparing (6) and (7) reveals the significance of product attributes in generating precise product titles.

5.4 Further Investigation

Cold-start Setting. The cold-start problem is a well-known issue in recommendation systems (Lee et al., 2019; Pan et al., 2019; Zhu et al., 2021). To assess our model's performance in a cold-start context, we have selected items from the testing sets that have not appeared in the training sets to construct the cold-start dataset for evaluation. For ID-based methods like CORE, we incorporate a "cold" token embedding into the item embeddings to supply prior knowledge, following the approach in (Li et al., 2023). The results are presented in Figure 4. It is evident that text-based methods significantly outperform ID-based approaches, primarily due to the limitations of randomly initialized cold-start item embeddings. Furthermore, our method surpasses Recformer, illustrating the effective incorporation of general knowledge and reasoning capabilities provided by LLMs.

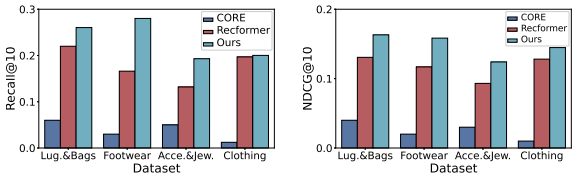

Figure 4: Performance comparison between our method with baselines in cold-start settings.

Zero-shot Setting. In this setting, the models are required to learn knowledge from pre-trained datasets and directly test on downstream datasets without further fine-tuning, thus ID-based methods are not applicable here. To ensure a fair comparison with Recformer, which undergoes pre-training on large-scale, recommendation specific datasets, we used models pre-trained on the footwear dataset to evaluate performance on three other datasets. We also employed a model trained on the luggage dataset to assess its performance on the footwear dataset. The superior performance given in Figure 5 demonstrates that our method can effectively capture and transfer learned knowledge to new tasks based on language understanding.

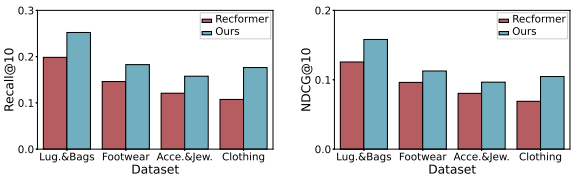

Figure 5: Performance comparison between our method with baselines in zero-shot settings.

Low Resource Setting. In this setting, we trained models on datasets with different ratios of training data. The experiment results are given in Figure 6. We can see that when the less training data is available, the text-based methods outperforms the ID-based CORE, this advantage stems from the transferable knowledge encoded in item texts. Additionally, as the amount of training data increases, our method shows a more significant performance improvement compared to Recformer, highlighting its efficiency in learning task-specific knowledge.

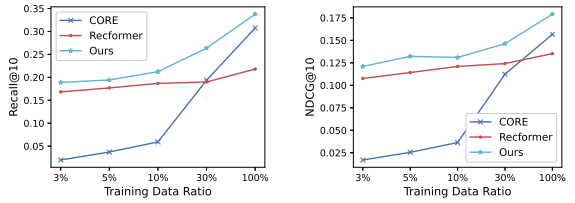

Figure 6: Comparison between our method with baselines in low-resource settings.

6 Conclusion

In this paper, we propose a sequential fashion recommendation system enhanced by a LLM. Our method consists of three stages: prompt creation, training and inference, and retrieval and ranking. First, we design specialized prompts that align the model with recommendation-specific goals. Second, we conduct efficient training to optimize the model. Third, we introduce a novel mix-up-based retrieval strategy that utilizes both ID and title embeddings to finalize item recommendations. Extensive experiments show our method significantly enhances fashion recommendation performance.

Limitations

Training and Inference Latency. Incorporating LLMs into recommendation systems introduces additional complexities in terms of time and space. Despite these challenges, the domain of enhancing LLM efficiency is evolving swiftly, presenting strategies to alleviate these concerns. For instance, parameter-efficient fine-tuning techniques can notably reduce memory requirements and training time. In terms of inference efficiency, there is a growing body of research dedicated to developing more efficient inference frameworks. Notable contributions include LLMLingua (Jiang et al., 2023), StreamingLLM (Xiao et al., 2023), and PagedAttention (Kwon et al., 2023). These innovations demonstrate the feasibility of reducing the time and space complexities of LLMs. Furthermore, considering the substantial performance improvements the LLM could bring, the increased complexity is a worthwhile investment.

Incorporating Visual Signals. Visual signals play an important role in shaping users' shopping decisions in the fashion domain. Our current approach focuses on textual data to model user interaction patterns, as incorporating images would significantly increase data collection and computational demands. However, integrating visual signals into our recommendation framework remains a promising direction. For instance, we could leverage multimodal LLMs to extract visual attributes such as color palette, lighting, textile type, shoulder style, and boot style (Zou et al., 2024). Incorporating these attributes into our LLM-based recommendation framework could enhance its effectiveness.

Security and Privacy Risks. Like other machine learning models, LLMs are vulnerable to various security and privacy risks (Liu et al., 2023; Chang et al., 2024; Liu et al., 2024). For instance, LLMs can exhibit memorization tendencies that make them susceptible to data extraction attacks, which may recover training samples and thereby compromise user privacy (Carlini et al., 2021). Addressing these risks with effective countermeasures is an important direction for future work.

References

Falcon llm. https://falconllm.tii.ae/falcon.html. Accessed: 2023-08-01.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Yuanhaur Chang, Han Liu, Evin Jaff, Chenyang Lu, and Ning Zhang. 2024. Sok: Security and privacy risks of medical ai. *arXiv preprint arXiv:2409.07415*.

Hai Chen, Fulan Qian, Jie Chen, Shu Zhao, and Yanping Zhang. 2021. Fg-rs: Capture user fine-grained preferences through attribute information for recommender systems. *Neurocomputing*, 458:195–203.

Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. 2019. Pog: personalized outfit generation for fashion recommendation at alibaba ifashion. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2662–2670.

Yashar Deldjoo, Fatemeh Nazary, Arnau Ramisa, Julian Mcauley, Giovanni Pellegrini, Alejandro Bellogin, and Tommaso Di Noia. 2022. A review of modern fashion recommender systems. *arXiv preprint arXiv:2202.02757*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming Zhu. 2020. Mamo: Memory-augmented meta-optimization for cold-start recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 688–697.

Koen van Gelder. Apparel, footwear and accessories retail e-commerce revenue in the united states from 2017 to 2027. Accessed: 2023-08-01.

Ruining He, Chunbin Lin, and Julian McAuley. 2016. Fashionista: A fashion-aware graphical system for exploring visually similar items. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 199–202.

Ruining He and Julian McAuley. 2016a. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE.

Ruining He and Julian McAuley. 2016b. Vbpr: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Min Hou, Le Wu, Enhong Chen, Zhi Li, Vincent W Zheng, and Qi Liu. 2019. Explainable fashion recommendation: A semantic attribute region guided approach. *arXiv preprint arXiv:1905.12862*.

Yupeng Hou, Binbin Hu, Zhiqiang Zhang, and Wayne Xin Zhao. 2022. Core: simple and effective session-based recommendation within consistent representation space. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*.

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*.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*.

Shuhui Jiang, Yue Wu, and Yun Fu. 2016. Deep bi-directional cross-triplet embedding for cross-domain clothing retrieval. In *Proceedings of the 24th ACM international conference on Multimedia*.

Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. 2017. Visually-aware fashion recommendation and design with generative image models. In *2017 IEEE international conference on data mining (ICDM)*, pages 207–216. IEEE.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*.

Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1073–1082.

Chenglin Li, Mingjun Zhao, Huanming Zhang, Chenyun Yu, Lei Cheng, Guoqiang Shu, Beibei Kong, and Di Niu. 2022. Recguru: Adversarial learning of generalized user representations for cross-domain recommendation. In *Proceedings of 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 Julian McAuley. 2023. Text is all you need: Learning language representations for sequential recommendation. *arXiv preprint arXiv:2305.13731*.

Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Improving outfit recommendation with co-supervision of fashion generation. In *The World Wide Web Conference*.

Han Liu, Yuhao Wu, Zhiyuan Yu, and Ning Zhang. 2024. Please tell me more: Privacy impact of explainability through the lens of membership inference attack. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 120–120. IEEE Computer Society.

Han Liu, Yuhao Wu, Shixuan Zhai, Bo Yuan, and Ning Zhang. 2023. Riatig: Reliable and imperceptible adversarial text-to-image generation with natural prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20585–20594.

Qiang Liu, Shu Wu, and Liang Wang. 2017. Deepstyle: Learning user preferences for visual recommendation. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval*, pages 841–844.

Siwei Liu, Iadh Ounis, Craig Macdonald, and Zaiqiao Meng. 2020. A heterogeneous graph neural model for cold-start recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 2029–2032.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Haixu Ma, Donglin Zeng, and Yufeng Liu. 2022. Learning individualized treatment rules with many treatments: A supervised clustering approach using adaptive fusion. *Advances in Neural Information Processing Systems*, 35:15956–15969.

Haixu Ma, Donglin Zeng, and Yufeng Liu. 2023. Learning optimal group-structured individualized treatment rules with many treatments. *Journal of Machine Learning Research*, 24(102):1–48.

Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 695–704.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820.

Zihua Si, Zhongxiang Sun, Xiao Zhang, Jun Xu, Xiaoxue Zang, Yang Song, Kun Gai, and Ji-Rong Wen. 2023. When search meets recommendation: Learning disentangled search representation for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1313–1323.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 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.

Huizhao Wang, Guanfeng Liu, Yan Zhao, Bolong Zheng, Pengpeng Zhao, and Kai Zheng. 2019. Dmfp: A dynamic multi-faceted fine-grained preference model for recommendation. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 608–617. IEEE.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Ruiping Yin, Kan Li, Jie Lu, and Guangquan Zhang. 2019. Enhancing fashion recommendation with visual compatibility relationship. In *The world wide web conference*, pages 3434–3440.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983.

Wenhui Yu, Xiangnan He, Jian Pei, Xu Chen, Li Xiong, Jinfei Liu, and Zheng Qin. 2021. Visually aware recommendation with aesthetic features. *The VLDB Journal*, 30:495–513.

Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In *Proceedings of the 12th ACM international conference on web search and data mining*.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38.

Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *CIKM*, pages 4653–4664. ACM.

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 Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Henry Peng Zou, Gavin Heqing Yu, Ziwei Fan, Dan Bu, Han Liu, Peng Dai, Dongmei Jia, and Cornelia Caragea. 2024. Eiven: Efficient implicit attribute value extraction using multimodal llm. *arXiv preprint arXiv:2404.08886*.