SOLAR: Serendipity Optimized Language Model Aligned for Recommendation

Anonymous ACL submission

Abstract

Recently, Large Language Models (LLMs) have shown strong potential in recommendation tasks due to their broad world knowledge and reasoning capabilities. However, applying them to serendipity-oriented recommendation remains challenging, mainly due to a domain gap of LLMs in modeling personalized user behavior and the scarcity of labeled serendipitous interactions. In this paper, we introduce **SOLAR** (**S**erendipity-**O**ptimized **L**anguage model **A**ligned for **R**ecommendation), a two-stage framework that addresses these challenges. To alleviate label scarcity, we adopt a weak supervision strategy: a sequential ID-based recommender generates candidate items, which are then reranked by an LLM acting as a preference judge to produce serendipity-aware pseudo-labels. To bridge the domain gap, we propose a domain-adaptive instruction tuning method (SUN) that aligns LLMs with recommendation tasks. Experiments on three real-world datasets show that **SOLAR** consistently improves both accuracy and serendipity over strong baselines, showing its effectiveness in enabling more diverse, user-centric recommendations. Code and dataset are released at https://github.com/SOLAR2025ARR/SOLAR.

1 Introduction

Recommender systems are essential tools for helping users discover items aligned with their interests. While existing recommendation algorithms (He et al., 2017; Kang and McAuley, 2018; Sun et al., 2019) focus on maximizing accuracy, they often reinforce users' past preferences by recommending similar items. This accuracy-centric design leads to the well-known *filter bubble* phenomenon (Pariser, 2011), where users are repeatedly exposed to homogeneous content, limiting exposure to novel or diverse items. Over time, this can result in content fatigue and reduced user engagement.

To address the filter bubble effect, the recommendation community has increasingly emphasized beyond-accuracy objectives such as diversity and serendipity (Ge et al., 2010; Díez et al., 2019). Serendipity, in particular, aims to surface recommendations that are not only relevant but also novel and pleasantly surprising (Kotkov et al., 2016). Various multi-objective optimization frameworks have been proposed to balance relevance with beyond-accuracy objectives (Li et al., 2021; Li and Tuzhilin, 2020). However, most existing methods still rely on heuristics and assumptions rather than real human feedback, due to the limited data for supervised training, which may not faithfully reflect human perceptions.

Recently, Large Language Models (LLMs) have emerged as promising alternatives due to their powerful semantic understanding and broad world knowledge (Lin et al., 2025). Unlike traditional collaborative filtering approaches that rely solely on user-item interactions, LLMs can directly reason over item semantics and user preferences in natural language (Sheng et al., 2024), enabling more flexible and nuanced recommendations. This raises the exciting possibility of using LLMs to deliver more serendipitous recommendations by capturing subtle user intents and surfacing less obvious but relevant content.

Despite this potential, LLMs face two key challenges in serendipity-oriented recommendation. First, a *domain gap* exists between general-purpose LLMs and recommendation tasks: LLMs excel at natural language understanding but lack collaborative filtering mechanisms, limiting their ability to model personalized preferences from interaction data (Zhao et al., 2024; Lin et al., 2023). Second, there is a significant *label scarcity* problem, as collecting high-quality serendipity annotations is costly and time-consuming (Fu et al., 2023; Kotkov et al., 2018). These issues hinder the direct application of LLMs to personalized, serendipity-aware recommendation scenarios.

To address these challenges, we propose **SOLAR** (**Serendipity-Optimized Language model Aligned for Recommendation**), a unified framework for building LLM-based recommenders optimized for serendipity. SOLAR contains two main components: (1) A human-aligned weak supervision pipeline, where a sequential ID-based recommender finetuned on limited human-labeled data generates candidate items, which are then reranked by an LLM acting as a preference judge to produce large-scale, serendipity-aware pseudo-labels; and (2) A domain-adaptive instruction tuning framework,

termed the recommendation-specialized unified tuning network (**SUN**), which aligns general-purpose LLMs with recommendation objectives using pseudo-labeled data and carefully designed prompts.

The contributions of our work are summarized as follows:

1. We propose **SOLAR**, a unified framework that aligns LLMs with human preference for generating accurate and serendipitous recommendations.

2. We introduce a human-aligned weak supervision approach that combines an ID-based recommender and an LLM-based reranker to generate large-scale, high-quality serendipity-aware pseudo-labels, mitigating the label scarcity problem.

3. We develop the SUN framework, which effectively bridges the domain gap between general-purpose LLMs and recommendation tasks via instruction tuning.

4. Extensive experiments on three real-world datasets demonstrate that **SOLAR** consistently improves both recommendation accuracy and serendipity, validating its potential in breaking filter bubbles and enhancing user satisfaction.

2 Related Works

2.1 LLMs for Recommender Systems

Recent advances in LLMs have opened up new opportunities for enhancing recommender systems, significantly extending their capabilities. Current LLM-based recommender approaches can broadly be categorized into generative methods (Liao et al., 2024; Geng et al., 2023; Zheng et al., 2024a; Lu et al., 2024; Bao et al., 2023), feature engineering techniques (Ren et al., 2024; Zhang et al., 2025; Du et al., 2023), and methods for improving representation learning (Rajput et al., 2023; Harte et al., 2023; Hou et al., 2023). Moreover, researchers have integrated LLMs into conversational recommendation scenarios (Yang et al., 2024; Gao et al., 2023; Feng et al., 2023) and interactive recommendation agents (Zhang et al., 2024b,a), aiming to align recommendation systems more closely with realistic user interactions.

Despite these advances, integrating LLMs into recommendation scenarios still faces significant challenges. One major issue is aligning LLM outputs with user preferences, especially given noisy and complex interaction data. Recent approaches have attempted to address this by combining LLMs with traditional recommendation models (Yue et al., 2023; Li et al., 2023; Wang et al., 2024) or finetuning LLMs using domain-specific instructions (Bao et al., 2023; Lu et al., 2024; Zhang et al., 2024c). Additionally, initial attempts to employ LLMs as rerankers have demonstrated their efficiency in existing recommendation pipelines (Hou et al., 2024). However, prior work typically still focus on accuracy objectives, indicating a critical gap for further research.

2.2 Serendipity in Recommender Systems

In real-world recommender systems, it's crucial to balance accuracy with beyond-accuracy objectives. One such objective is serendipity (Kotkov et al., 2016). In the Merriam-Webster dictionary, serendipity is explained as: *the faculty or phenomenon of finding valuable or agreeable things not sought for*. While there is no agreement on how to define serendipity quantitatively yet due to its subjective nature, most definition agree on its two key elements: **relevance** and **unexpectedness**.

Early research mainly focus on building serendipity-oriented recommender systems through multi-objective optimization (Díez et al., 2019; Li and Tuzhilin, 2020; Li et al., 2021). Learning to optimize multiple metrics helps the system provide relevant recommendations while catering to user satisfaction, thus enhancing user engagement (Rodriguez et al., 2012). However, due to the scarcity of serendipity-labeled data, prior work often relies on heuristics or assumptions to guide model design or preprocessing, which may fail to capture the nuances of human preferences.

Recent advances in LLMs, particularly using LLMs as preference judges (Li et al., 2025; Gu et al., 2025), offer a promising solution to this problem. In our work, we propose a weak supervision strategy that leverages a powerful LLM to rerank candidate recommendations generated by an ID-based sequential model. This two-stage pipeline produces serendipity-aware pseudo-labels, enabling scalable alignment of LLMs with human-centric recommendation objectives.

3 Methodology

In this section, we introduce **SOLAR**, a framework for building serendipity-oriented recommender system based on LLMs. SOLAR addresses two key challenges in applying LLMs to serendipitous recommendation: the *domain gap* between language modeling and collaborative filtering, and the *scarcity* of human-labeled serendipity data.

As illustrated in Figure 1, SOLAR consists of three main stages:

1. **ID-based Model Training.** We first train an ID-based sequential recommender on large-scale relevance data, and then finetune it using limited serendipity-labeled interactions. This yields an initial model capable of generating serendipity-aware recommendation candidates.

2. **LLM-based Reranking.** Given the candidate items from the ID-based model, we employ an LLM as a preference judge to rerank the recommendations based on serendipity-oriented prompts. This weak supervision pipeline allows us to construct large-scale pseudo-labeled data with human-aligned serendipity signals.

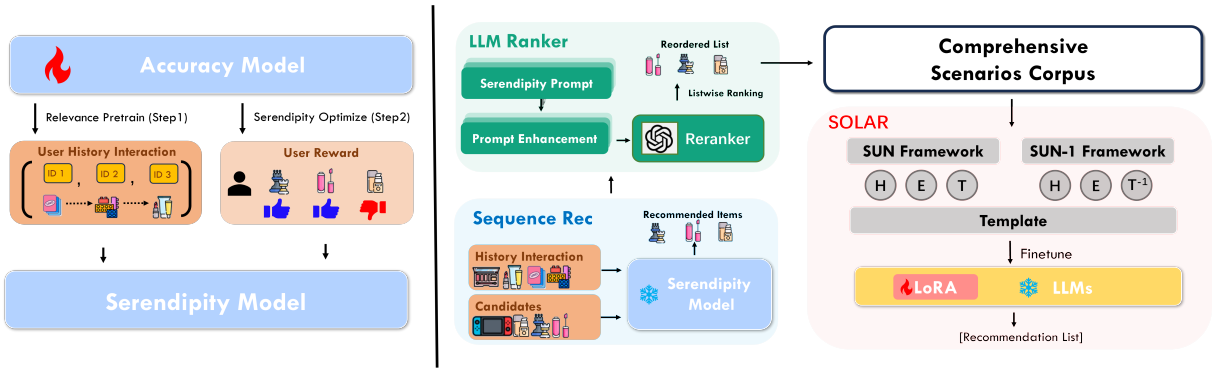

Figure 1: Overview of the **SOLAR** framework. It consists of three stages: (1) training an ID-based recommender with accuracy and serendipity objectives (left), (2) reranking its outputs using an LLM guided by serendipity prompts to generate pseudo-labeled data (mid), and (3) instruction-tuning a general-purpose LLM via the SUN framework for serendipitous recommendation (right).

3. **Domain Adaptation via SUN.** We convert the reranked results into structured instruction-style data by encoding user history, recommendations and serendipity explanations. We then finetune a general-purpose LLM using the proposed **SUN** (Specialized Unified Tuning Network) framework, which aligns the LLM with serendipity-aware recommendation objectives.

We next formalize the sequential recommendation problem and provide detailed descriptions of each stage in the SOLAR framework.

3.1 Problem Formulation

To begin with, we formalize the sequential recommendation problem. For each user u, we observe a historical interaction sequence $H = \{h_1^u, h_2^u, \ldots, h_n^u\}$, representing items the user interacted with until time step n. Given a user set $U = \{u_1, u_2, \ldots, u_{|U|}\}$ and an item set $I = \{i_1, i_2, \ldots, i_{|I|}\}$, our goal is to predict the next item h_{n+1}^u that user u is likely to interact with. Formally, we aim to estimate the conditional probability:

$$\mathbb{P}(i \mid H, \boldsymbol{\theta}), \quad \forall i \in I$$

Then, the next predicted item is selected as:

$$h_{n+1}^u = \arg\max_{i \in I} \mathbb{P}(i \mid H, \boldsymbol{\theta}),$$

where $\boldsymbol{\theta}$ denotes the parameters of the recommendation model.

3.2 ID-based Model Training

Consider a set of users U of size N and items I of size M. Each user u_j has an interaction history $H_j = \{i_{k_1}, i_{k_2}, \ldots, i_{k_l}\}$. Given model parameters $\boldsymbol{\theta}$, we predict the probability that user u_j interacts with item i_k:

$$p_{jk}(\boldsymbol{\theta}) = p(i_k \mid u_j, \boldsymbol{\theta}).$$

We define two types of binary labels: relevance r_{jk} and serendipity s_{jk}:

$$r_{jk} = \begin{cases} 1 & \text{if } u_j \text{ considers } i_k \text{ relevant} \\ 0 & \text{otherwise} \end{cases}$$

$$s_{jk} = \begin{cases} 1 & \text{if } u_j \text{ considers } i_k \text{ serendipitous} \\ 0 & \text{otherwise} \end{cases}$$

Ideally, to balance accuracy and serendipity, we would jointly optimize a combined loss:

$$\mathcal{L}(\boldsymbol{\theta}) = (1-\lambda)\mathcal{L}(\mathbb{P}^r(\boldsymbol{\theta})) + \lambda\,\mathcal{L}(\mathbb{P}^s(\boldsymbol{\theta})),$$

where λ is a hyperparameter balancing both objectives:

$$\mathcal{L}(\mathbb{P}^r(\boldsymbol{\theta})) = -\sum_{j,k} \Big(r_{jk} \log p_{jk}^r(\boldsymbol{\theta}) + (1 - r_{jk}) \log(1 - p_{jk}^r(\boldsymbol{\theta})) \Big),$$

$$\mathcal{L}(\mathbb{P}^s(\boldsymbol{\theta})) = -\sum_{j,k} \Big(s_{jk} \log p_{jk}^s(\boldsymbol{\theta}) + (1 - s_{jk}) \log\big(1 - p_{jk}^s(\boldsymbol{\theta})\big) \Big),$$

However, due to the extreme scarcity of serendipity-labeled data, directly training the model with this joint objective is not feasible. To overcome this limitation, we adopt a two-stage transfer learning approach inspired by (Pandey et al., 2018), as illustrated in Figure 2:

1. **Relevance Pre-training:** Initially, we optimize only the relevance objective using a large-scale relevance-labeled dataset.

2. **Serendipity-aware Finetuning:** Subsequently, we freeze all parameters except the final dense layer and finetune this layer with the limited

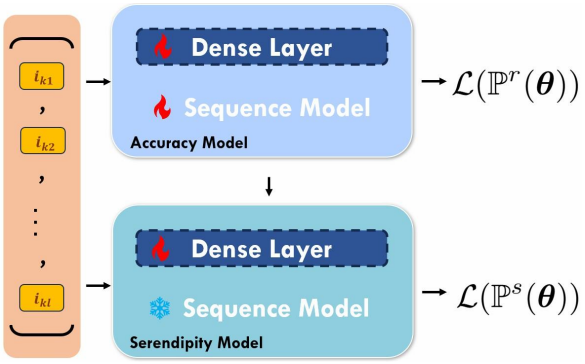

Figure 2: ID-based Model Training. Due to limited serendipity labels, we first optimize the model using relevance-labeled data, then finetune the final layer using scarce serendipity annotations.

serendipity-labeled dataset to introduce serendipity signals into the model[1].

This transfer learning strategy allows us to incorporate serendipity signals despite limited annotations, yielding a unified recommendation model capable of balancing accuracy and serendipity.

3.3 LLM-based Reranking and Serendipity Refinement

Due to the scarcity of serendipity-labeled data, ID-based recommenders, even after finetuning, may still fail to capture nuanced human perceptions of serendipity. To address this limitation, we incorporate a powerful large language model (e.g., GPT-4 (Achiam et al., 2023)) as a preference judge to rerank initial recommendations, effectively injecting serendipity awareness through a weak supervision process.

As shown in Figure 3, our prompting strategy consists of two stages:

Serendipity Understanding via In-Context Learning. We begin by prompting the LLM with a role definition and background context to establish a general understanding of serendipity. Then, we provide labeled examples of user-item interactions annotated with serendipity scores, including both highly serendipitous and non-serendipitous cases, to guide the LLM through in-context learning. Based on these examples, the LLM is asked to generate an explanation of its learned serendipity criteria, which is used to guide the reranking process.

Candidate List Reranking. With the acquired serendipity understanding and the user's interaction history, the LLM is prompted to rerank the candidate items produced by the ID-based recommender. For each item, the model provides a concise explanation, implicitly integrating unexpectedness and relevance to produce a refined recommendation list aligned with human notions of serendipity.

[1]Human annotation data on serendipity is scarce within the community. We have collected all publicly available labels to date.

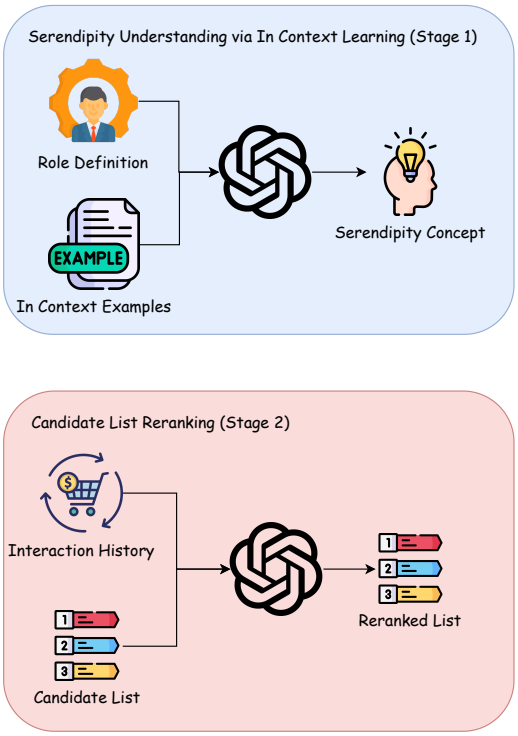

Figure 3: LLM-based reranking and serendipity refinement. Given a user's interaction history and an initial recommendation list from the ID-based model, the LLM assesses and reranks the items to produce a serendipity-enhanced recommendation list.

The reranked outputs, along with their corresponding explanations, are then used to construct instruction-style training data for downstream domain adaptation.

3.4 Domain Adaptation

To align large language models, which operate in the natural language domain, with the goals of serendipitous recommendation, we transform recommendation tasks into instruction-style prompts that LLMs can understand and complete.

3.4.1 Key elements in instruction: the SUN Framework

To bridge the collaborative filtering and language modeling domains, we propose **SUN** (Recommendation Specialized Unified Tuning Network), a unified instruction tuning framework. SUN encodes essential elements of recommendation tasks into natural language instructions, enabling effective adaptation of general-purpose LLMs. We first describe the overall framework briefly, and introduce each component in detail with illustrative examples.

Formally, the framework is defined by the following pair of triplets:

$$\mathbf{SUN} = (\mathbf{H}, \mathbf{E}, \mathbf{T}) \quad \text{and} \quad \mathbf{SUN}^{-1} = (\mathbf{H}, \mathbf{E}, \mathbf{T}^{-1})$$

Table 1: Prompt examples of the SUN (forward tasks) and SUN^{-1} (inverse tasks) instruction templates. We highlight three key components: History, Engagement, and Task.

Category	Example
$\langle H, \_, T_0 \rangle$	Given the user's interaction history: `{history}`, what is the optimal product to suggest next ?
$\langle H, P, T_1 \rangle$	Please recommend an item to the user based on the following information about the user: `{history}`, the user's historical interaction, which is as follows: `{preference}`. Try to recommend one item from the following candidates that is consistent with the user's preference: `{candidate_items}`.
$\langle H, P, T_2 \rangle$	You have some information about this user, which is shown below: `{preference}`, the user's historical interactions: `{history}`. Please recommend the reranking order of items for the user from the following candidates: `{candidate_items}`.
$\langle H, I, T_3 \rangle$	The user enjoys being surprised and has shown implicit preferences based on their interactions: `{history}`. The user's current intention may be vague as follows: `{intention}`. Based on this information, evaluate the following candidate item: `{candidate_item}`, please answer "Yes" or "No" for the fitness of candidate.
$\langle H, \_, T_3^{-1} \rangle$	Given the following historical interactions of the user: `{history}`, and the next recommended item: `{item}`. Please infer the specific intention that would likely lead to this recommendation.

where the **H**, **E**, **T** stands for **History**, **Engagement Profile** and **Task** respectively. While **SUN** instructs the LLMs to perform recommendation tasks given **History** and **Engagement Profile**, **SUN^{-1}**, which serves as a task enrichment with input inversion, instructs the LLMs to infer **Engagement Profile** given **Task** output and **History**.

History (H) represents users' historical information, such as interaction history. In our scenario, we use a sequence of most recently interacted items to represent history.

Engagement Profile (E) provides explanation about user's preference pattern, which is a key element used for modeling and interpreting user behavior. The engagement profile can include two kinds of information: **preferences (P)**, which indicates user's explicit preference, and **intentions (I)**, which represents user's implicit preference.

Task (T) denotes the type of recommendation task. We define four task types, denoted as T_0 to T_3:

- *Generative Recommendation* (T_0): The model directly generates recommendations rather than selecting from existing candidates. This enables it to produce novel outputs based on the user's history and engagement profile.
- *Direct Recommendation* (T_1): The model selects the most suitable item from a predefined set of candidates, identifying the best match without generating new items.
- *Reranking* (T_2): Given a set of candidate items, the model reorders them according to a specific objective, such as optimizing accuracy or serendipity.
- *Matching* (T_3): The model determines whether a given candidate item matches the user's preferences or intent, producing a binary decision ("Yes" or "No").

Table 1 presents illustrative examples of instructions described by the above mentioned framework. Complete templates are provided in Appendix G.

3.4.2 Engagement Profile Generation

While **H** and **T** can be directly obtained from the dataset or previous steps, the **E** component, describing users' implicit and explicit preferences, is typically unavailable. To address this, we prompt a teacher LLM (e.g., GPT-4) to infer the engagement profile from historical interactions. This inferred profile provides richer semantic grounding and helps contextualize the recommendation task.

Prompt for Engagement Profile Generation

User's historical interactions: `{interaction}`. Based on these movie titles, use your knowledge to generate a description of the user's implicit preferences, such as their favorite genres, themes, or notable patterns. `{constraint}`

Figure 4: An example prompt for generating engagement profile from movie interaction data.

Figure 4 gives an example of the prompt used for engagement profile generation. Given the user's historical interactions, the teacher LLM is prompted to provide an explanation that forms the user's engagement profile. Moreover, to address different real-world scenarios, we ask the teacher LLM to generate engagement profiles reflecting varying degrees of preference clarity, including both implicit (**I**) and explicit (**P**) preferences. Additional details about the engagement profile are provided in Appendix D and H.

3.4.3 Instruction Finetuning

We train our large language model for serendipitous recommendations using instruction finetuning (Zhang et al., 2024d). Specifically, we adopt LLaMA (Touvron et al., 2023) as the backbone and finetune it on the instruction dataset generated with the SUN framework.

For supervised finetuning, we optimize the model parameters θ by minimizing the following loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{|D|} \sum_{(x_i, y_i) \in D} \log p_\theta(y_i | x_i) \quad (1)$$

Here, D is our instruction dataset, where each x_i (including the user's historical interactions, engagement profile, and task prompt) is paired with the target serendipitous recommendation y_i.

4 Experiments

In this section, we first provide our experimental setup, and then present the results as well as analyses of our proposed approach. Additional implementation details are presented in Appendix B. To evaluate the real world effectiveness of our approach, we also conduct an offline A/B test, whose details are provided in Appendix F.

We conduct experiments to answer the following research questions:

- **RQ1:** How does **SOLAR** perform compared with baseline models?
- **RQ2:** How good is the quality of the generated pseudo-labeled data?
- **RQ3:** How does the amount of instruction data affect SOLAR's domain adaptation performance?
- **RQ4:** How important are **SOLAR**'s components?

4.1 Experimental Setup

Table 2: Dataset Statistics.

Dataset	# Users	# Items	# Actions	Sparsity
MovieLens	10,684	11,544	1.05M	99.15%
Books	152,776	65,631	2.94M	99.97%
Movies&TV	69,993	27,560	0.69M	99.96%

4.1.1 Datasets

We conduct experiments on three real-world datasets from different domains: MovieLens (Harper and Konstan, 2015; Kotkov et al., 2018), Amazon Books (Ni et al., 2019; Fu et al., 2023), and Amazon Movies&TV (Ni et al., 2019; Fu et al., 2023). These datasets provide distinct item titles and text-based reviews for large-scale relevance-labeled datasets with real user-labeled serendipity data for a small subset of users, we found that **these datasets are currently the only publicly available ones with user-labeled serendipity data**. Table 2 presents a summary of dataset statistics.

4.1.2 Evaluation Metrics

Following prior work on serendipitous sequential recommendation (Fu et al., 2023), We adopt a user-level train/test split strategy. To accommodate the context length limitations of LLMs, each positive item is evaluated against 19 randomly sampled negatives.

For accuracy evaluation, we report **Hit Rate at 1 (HR@1)**. For serendipity evaluation, we use **HR_{seren}@1**, which measures the hit rate of items labeled as serendipitous by users.

4.1.3 Baselines

We compare **SOLAR** against a diverse set of representative baselines spanning different paradigms of recommendation systems:

- *Traditional sequential recommenders:* These include **Caser** (Tang and Wang, 2018), **SASRec** (Kang and McAuley, 2018), **BERT4Rec** (Sun et al., 2019), and **S^3-Rec** (Zhou et al., 2020), which model user behavior sequences using CNNs, self-attention, or mutual information maximization.
- *LLM-based recommenders:* This category covers models that incorporate large language models in various forms, including **TALLRec** (Bao et al., 2023), **P5** (Geng et al., 2023), **LLMRank** (Hou et al., 2024), **RecLM** (Lu et al., 2024). These models leverage pretrained language models with prompt engineering or finetuning for recommendation tasks.
- *General-purpose LLM:* We also include **GPT-4o** (Achiam et al., 2023), a state-of-the-art instruction-following model, as a zero-shot recommender to assess the capability of off-the-shelf LLMs in recommendation scenarios.

Additional details on baselines are provided in Appendix C.

4.2 Performance Comparison (RQ1)

Performance comparison of SOLAR against baselines is summarized in Table 3.

Among ID-based models, method that incorporate item attribute information (S^3-Rec) outperform models that rely solely on collaborative signals (Caser, BERT4Rec, and SASRec). While ID-based models excel in capturing collaborative patterns, leading to strong results on accuracy metrics, they fall short in generating serendipitous recommendations, likely due to the inherent sparsity of such interactions in training data.

LLM-based methods, on the other hand, benefit from strong language understanding capabilities. However, even with some degree of domain adaptation, they lack the ability to fully leverage collaborative filtering signals. This results in subpar accuracy performance compared to ID-based methods. Nevertheless, LLM-based approaches generally outperform ID-based ones on serendipity metrics, possibly owing to their superior capacity to model subjective human preferences.

Our proposed SOLAR framework consistently achieves comparable or superior performance across most datasets in terms of both accuracy and serendipity.

Table 3: Comparison of **SOLAR** and baselines on the MovieLens, Movies&TV, and Books datasets in terms of HR@1 and HR_{seren}@1. The best and second-best results are in bold and underlined, respectively.

Dataset	Metric	SASRec	BERT4Rec	S^3-Rec	Caser	P5	TALLRec	LLMRank	RecLM	GPT-4o	SOLAR
Movielens	HR@1	0.1936	0.1616	0.2055	0.1985	0.0234	0.0310	0.0603	0.1353	0.1853	**0.2160**
	HR_{seren}@1	0.0568	0.0341	0.0472	0.0251	0.0138	0.0141	0.0219	0.0894	0.0395	**0.1284**
Movies&TV	HR@1	0.1478	0.1369	0.1417	0.1387	0.0398	0.0341	0.0584	**0.1591**	0.1256	0.1451
	HR_{seren}@1	0.0181	0.0084	0.0325	0.0253	0.0118	0.0103	0.1207	0.1131	0.0443	**0.1314**
Books	HR@1	0.1383	0.1304	**0.1410**	0.1391	0.0323	0.0385	0.0537	0.1012	0.1097	0.1203
	HR_{seren}@1	0.0194	0.0146	0.0250	0.0146	0.0089	0.0149	0.0187	0.0719	0.0496	**0.0902**

This improvement stems from our strategy to address the label scarcity and domain gap problems. With abundant serendipity-aware labeled data and effective domain adaptation to integrating collaborative signals into LLMs, SOLAR is capable of providing both accuracy and serendipitous recommendations.

4.3 Pseudo-Label Quality (RQ2)

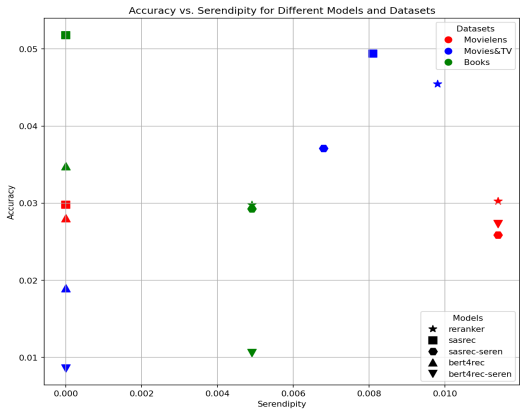

Figure 5: Trade-off between accuracy (**HR@1**) and serendipity ($\mathbf{HR}_{seren}$**@1**) across recommendation models and datasets. The star denotes our hybrid model.

To assess the quality of pseudo-labels used for instruction tuning, we evaluate our hybrid model, which combines a serendipity-tuned SASRec with an LLM-based reranker. We compare it against four baselines: **SASRec**, **BERT4Rec**, and their serendipity-tuned variants.

As shown in Table 4, the hybrid model achieves the best or competitive performance across both accuracy and serendipity on all datasets. It notably outperforms others on MovieLens and Movies&TV, and provides the highest serendipity on Books.

Figure 5 further illustrates the trade-off between accuracy and serendipity. While baseline models typically favor one metric at the cost of the other, our hybrid approach gets the best of both worlds (on the upper-right parts for each dataset).

These results demonstrate that the hybrid model can effectively approximate user preferences, providing high-quality pseudo-labels for downstream training.

Table 4: Comparison of our hybrid approach against baseline models (**SASRec**, **BERT4Rec**, and their serendipity-tuned variants) on **HR@1** and $\mathbf{HR}_{seren}$**@1** under **full ranking** setting.

Model	Movielens		Movies&TV		Books	
	Acc	Seren	Acc	Seren	Acc	Seren
SASRec	0.0298	0.0000	**0.0494**	0.0081	0.0518	0.0000
SASRec-seren	0.0259	0.0114	0.0371	0.0068	0.0293	0.0049
BERT4Rec	0.0281	0.0000	0.0190	0.0000	**0.0348**	0.0000
BERT4Rec-seren	0.0273	0.0114	0.0086	0.0000	0.0106	0.0049
Our reranker	**0.0303**	**0.0114**	0.0455	**0.0098**	0.0298	**0.0049**

4.4 Impact of Instruction Data Scale (RQ3)

To evaluate the scalability of our domain adaptation approach, we examine how different proportions of instruction data affect model performance. We sample four augmentation ratios: 0 (unfinetuned LLaMA), 0.33, 0.66, and 1.0, representing increasing amounts of instruction data relative to the full training set.

As shown in Table 5, both accuracy and serendipity improve as more instruction data is used. The gains are especially pronounced in serendipity, suggesting that the domain-adaptive tuning benefits most from serendipity-aware signals. However, performance gains begin to plateau near full data usage, indicating diminishing returns.

These results confirm that SOLAR's instruction tuning scales effectively with data and contributes to better alignment with user-centric recommendation goals.

Table 5: Proportion control of **SOLAR** on different data augmentation ratios.

Model	Movielens		Movies&TV		Books	
	Acc	Seren	Acc	Seren	Acc	Seren
UnfinetunedLlama (0)	0.1284	0.0675	0.0978	0.0403	0.1214	0.0317
SOLAR (0.33)	0.1651	0.0953	0.1154	0.0949	0.1147	0.0538
SOLAR (0.66)	0.1944	0.1156	0.1349	0.1166	0.1184	0.0748
SOLAR	**0.2160**	**0.1284**	**0.1451**	**0.1314**	**0.1203**	**0.0902**

4.5 Ablation Study (RQ4)

To assess the contribution of each component in **SOLAR**, we conduct an ablation study by evaluating the following variants:

- **w/o reranker:** Removes the LLM reranker and

Table 6: Comparison of **SOLAR** and its ablated variants on three datasets. Metrics reported: HR@1 and HR_{seren}@1. The best and second-best results are in bold and underlined, respectively.

Dataset	Metric	SOLAR	w/o reranker	w/o SM	w/o SM & reranker	w/o SUN	NoAugment
Movielens	HR@1	**0.2160**	0.1524	<u>0.1828</u>	0.1386	0.0363	0.1354
	HR_{seren}@1	**0.1284**	0.1101	<u>0.1193</u>	0.0780	0.0398	0.0734
Movies&TV	HR@1	**0.1451**	0.0915	<u>0.1072</u>	0.0878	0.0413	0.0928
	HR_{seren}@1	**0.1314**	0.0949	<u>0.1168</u>	0.0584	0.0487	0.0401
Books	HR@1	**0.1203**	0.0962	0.0909	0.1077	0.0527	<u>0.1123</u>
	HR_{seren}@1	**0.0902**	0.0598	<u>0.0824</u>	0.0255	0.0312	0.0333

uses only the serendipity-tuned ID model for label generation.

- **w/o SM:** Replaces the serendipity-tuned ID model with a standard accuracy-optimized sequential model.
- **w/o SM & reranker:** Removes both the reranker and the serendipity-tuned model.
- **w/o SUN:** Retaining only the sequential recommendation task without **SUN** framework.
- **NoAugment:** Trains only on limited human-labeled data without pseudo-labels.

Results in Table 6 show that removing any component leads to noticeable drops in both accuracy and serendipity. The LLM reranker and the serendipity-tuned model each contribute to performance gains, with the reranker having a more pronounced impact on serendipity. Removing both results in substantial degradation. Among all components, the SUN framework is the most critical, as its removal leads to the largest overall decline, showing its central role in aligning collaborative signals with language-based reasoning.

These results confirm that each component of **SOLAR** plays a complementary role, and their integration is essential for achieving both accurate and serendipitous recommendations.

4.6 Case Study

Table 7 presents a real-world case study illustrating how recommendations can be generated based on a user's historical viewing history. In this scenario, the user's previously watched films (e.g., The Witch, Sea Inside) reflect a preference for works characterized by rich storytelling, emotional depth, and socially relevant themes. Given a set of candidate movies (e.g., 7 Women, Without Bias) including target item (e.g., Remember), each model (**GPT4o**, **RecLM**, and **SOLAR**) was tasked with selecting five recommended items from candidate items. By comparing the resulting recommendation lists. Table 7 highlights key differences in recommendation approaches. GPT4o overexplains, burying recommendations in excessive detail.

Table 7: Comparison of recommendation lists generated by **GPT4o**, **RecLM**, and our **SOLAR** method. Each model selects items based on the user's historical interactions and candidate pool. Compared to **GPT4o** and **RecLM**, **SOLAR** provides a more aligned and diverse set of recommendations, effectively balancing accuracy and serendipity to yield the best overall results. We highlight texts for history items, candidate items, target item and unpredicted item.

Input Example	
Instruction	You are an expert recommender engine. You need to select a recommendation list considering user's historical interactions. The historical interactions are provided as follows: The Witch, Sea Inside, Long Way Round, Project Nim, Lady Bird, Best of Youth, La meglio gioventù, Take Me Home, The Hunting Ground. The candidate items are: Idle Class, The, 7 Women, Great Santini, The, Love(Szerelem), Shadow of the Vampire, We Are the Best! (Vi är bäst!), April 9th, Remember, Without Bias, Elvira, Mistress of the Dark. Please select a recommendation list with 5 different items from candidate items.
GPT4o	To generate a recommendation list, we can consider the user's historical interactions to identify patterns in their preferences. Based on the provided historical data, the user appears to enjoy movies that are critically acclaimed, have emotional depth, explore human experiences, or involve impactful storytelling. Using these preferences, here's a curated list of recommendations: Love(Szerelem), Shadow of the Vampire, Idle Class, The, Remember, Without Bias.
RecLM	7 Women, Without Bias, Remember, Elvira, Mistress of the Dark, Hurricane of Fun
SOLAR	Remember, Love (Szerelem), Great Santini, The, We Are the Best! (Vi är bäst!), 7 Women

RecLM hallucinates, suggesting movies outside the candidate set. In contrast, SOLAR effectively and accurately identifying the target item with a concise recommendation list that surpasses the other two models.

5 Conclusion

We propose **SOLAR**, a serendipity-optimized recommendation framework built upon large language models, which addresses two key challenges in serendipitous recommendation: the domain gap between language modeling and user behavior modeling, and the scarcity of serendipity-labeled data. To mitigate label scarcity, SOLAR leverages a weak supervision strategy that combines a serendipity-tuned ID-based model with an LLM-based reranker to generate high-quality pseudo-labels. To bridge the domain gap, we introduce the **SUN** framework, which aligns LLMs with collaborative filtering signals through domain-adaptive instruction tuning. Experiments on three real-world datasets demonstrate that SOLAR consistently outperforms strong baselines in both accuracy and serendipity. These results show the effectiveness of aligning LLMs with serendipitous recommendation objectives through weak supervision and instruction tuning, offering a scalable path toward more diverse and user-aligned recommendation systems.

Limitations

Although **SOLAR** demonstrates improved accuracy and serendipity, it still faces several potential risks and limitations. A potential risk lies in popularity bias, where the model may favor frequently appearing items in training data, though this could be mitigated through diversity-aware sampling strategies. Furthermore, using LLMs for large-scale recommendation tasks may face efficiency challenges. In this work, in our evaluation setting the input size is limited, and thus, noticeable efficiency issues do not arise. However, this constraint may limit its applicability in computationally restricted settings. We conduct experiments to evaluate efficiency, and the results can be found in the Appendix E. Future work may includes exploration on improving efficiency and scalability.

Ethics Statement

Our **SOLAR** framework aims to enhance recommendation diversity while maintaining user privacy and fairness. We rely on anonymized historical data and adhere to data protection standards. While serendipity may influence user preferences, we will strive to avoid biases and harmful content. Ongoing monitoring, transparency about recommendation processes, and allowing users to adjust or opt out of personalized suggestions help ensure ethical and responsible deployment.

References

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

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*, RecSys '23, page 1007–1014, New York, NY, USA. Association for Computing Machinery.

Jorge Díez, David Martínez-Rego, Amparo Alonso-Betanzos, Oscar Luaces, and Antonio Bahamonde. 2019. Optimizing novelty and diversity in recommendations. *Progress in Artificial Intelligence*, 8:101–109.

Yingpeng Du, Di Luo, Rui Yan, Hongzhi Liu, Yang Song, Hengshu Zhu, and Jie Zhang. 2023. Enhancing job recommendation through llm-based generative adversarial networks. *Preprint*, arXiv:2307.10747.

Yue Feng, Shuchang Liu, Zhenghai Xue, Qingpeng Cai, Lantao Hu, Peng Jiang, Kun Gai, and Fei Sun. 2023. A large language model enhanced conversational recommender system. *arXiv preprint arXiv:2308.06212*.

Zhe Fu, Xi Niu, and Li Yu. 2023. Wisdom of crowds and fine-grained learning for serendipity recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 739–748.

Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *Preprint*, arXiv:2303.14524.

Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 257–260.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2023. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). *Preprint*, arXiv:2203.13366.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

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

Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 1096–1102. ACM.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2023. Learning vector-quantized item representation for transferable sequential recommenders. *Preprint*, arXiv:2210.12316.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, pages 364–381. Springer.

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

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

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Denis Kotkov, Joseph A Konstan, Qian Zhao, and Jari Veijalainen. 2018. Investigating serendipity in recommender systems based on real user feedback. In *Proceedings of the 33rd annual acm symposium on applied computing*, pages 1341–1350.

Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. 2016. A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111:180–192.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *Preprint*, arXiv:2411.16594.

Pan Li, Maofei Que, Zhichao Jiang, Yao Hu, and Alexander Tuzhilin. 2021. Purs: Personalized unexpected recommender system for improving user satisfaction. *Preprint*, arXiv:2106.02771.

Pan Li and Alexander Tuzhilin. 2020. Latent unexpected recommendations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(6):1–25.

Xiangyang Li, Bo Chen, Lu Hou, and Ruiming Tang. 2023. Ctrl: Connect collaborative and language model for ctr prediction. *arXiv preprint arXiv:2306.02841*.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. *Preprint*, arXiv:2312.02445.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2025. How can recommender systems benefit from large language models: A survey. *ACM Trans. Inf. Syst.*, 43(2).

Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2023. A multi-facet paradigm to bridge large language model and recommendation. *arXiv preprint arXiv:2310.06491*.

Wensheng Lu, Jianxun Lian, Wei Zhang, Guanghua Li, Mingyang Zhou, Hao Liao, and Xing Xie. 2024. Aligning large language models for controllable recommendations. *Preprint*, arXiv:2403.05063.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Gaurav Pandey, Denis Kotkov, and Alexander Semenov. 2018. Recommending serendipitous items using transfer learning. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1771–1774, New York, NY, USA. Association for Computing Machinery.

Eli Pariser. 2011. *The Filter Bubble: What the Internet Is Hiding from You.*

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender systems with generative retrieval. *Preprint*, arXiv:2305.05065.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 3464–3475. ACM.

Mario Rodriguez, Christian Posse, and Ethan Zhang. 2012. Multiple objective optimization in recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 11–18.

Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua. 2024. Language models encode collaborative signals in recommendation. *arXiv preprint arXiv:2407.05441*.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Jianling Wang, Haokai Lu, Yifan Liu, He Ma, Yueqi Wang, Yang Gu, Shuzhou Zhang, Shuchao Bi, Lexi Baugher, Ed Chi, et al. 2024. Llms for user interest exploration: A hybrid approach. *arXiv preprint arXiv:2405.16363*.

Li Yang, Anushya Subbiah, Hardik Patel, Judith Yue Li, Yanwei Song, Reza Mirghaderi, and Vikram Aggarwal. 2024. Item-language model for conversational recommendation. *Preprint*, arXiv:2406.02844.

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. *arXiv preprint arXiv:2311.02089*.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024a. On generative agents in recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1807–1817.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024b. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM on Web Conference 2024*, pages 3679–3689.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024c. Recommendation as instruction following: A large language model empowered recommendation approach. *ACM Trans. Inf. Syst.* Just Accepted.

Kaike Zhang, Qi Cao, Yunfan Wu, Fei Sun, Huawei Shen, and Xueqi Cheng. 2025. Lorec: Large language model for robust sequential recommendation against poisoning attacks. *Preprint*, arXiv:2401.17723.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024d. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.

Hongke Zhao, Songming Zheng, Likang Wu, Bowen Yu, and Jing Wang. 2024. Lane: Logic alignment of non-tuning large language models and online recommendation systems for explainable reason generation. *arXiv preprint arXiv:2407.02833*.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024a. Adapting large language models by integrating collaborative semantics for recommendation. *Preprint*, arXiv:2311.09049.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024b. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

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.

A Dataset and Preprocessing

1. **MovieLens:** A classic movie recommendation dataset (Harper and Konstan, 2015) for movie recommendation systems, which comprises user ratings on movies and comprehensive textual descriptions of movies. The dataset also includes serendipity labels for a subset of movie reviews, obtained from the Serendipity 2018 dataset (Kotkov et al., 2018).

2. **Books:** A large-scale dataset derived from the *Amazon Review* dataset (Ni et al., 2019) containing millions of book metadata entries, user reviews, and ratings. It is commonly used for research on personalized recommendation systems. The dataset also includes serendipity labels for a subset of book reviews, obtained from the SerenLens dataset (Fu et al., 2023).

3. **Movies&TV:** A dataset also sourced from *Amazon Review*, including a wide range of movies and TV shows, including titles, genres, release dates, and millions of user reviews and ratings. It provides an essential benchmark for testing recommendation algorithms and analyzing trends in user preferences. The dataset also includes serendipity labels for a subset of movie and TV reviews, obtained from the SerenLens dataset.

We preprocess each dataset as follows:

1. **MovieLens** (Serendipity-2018): keep only ratings after June 1, 2017; remove users and items with fewer than 5 interactions.

2. **Books** (Amazon Reviews 2014): keep only reviews after January 1, 2012; remove users and items with fewer than 10 interactions.

3. **Movies&TV** (Amazon Reviews 2014): keep only reviews after January 1, 2012; remove users and items with fewer than 5 interactions.

For each dataset, we randomly split users into 80% for training and 20% for testing. For the training users, we further use the last interaction of each user sequence as the validation set, and the remaining interactions as the training data.

The datasets used in this study are publicly available and widely used in academic research. To the best of our knowledge, they do not contain personally identifiable information (PII) or offensive content. We rely on the dataset providers' documentation and our own checks to ensure the ethical use of data.

B Implementation Details

We implement our SOLAR framework with the following configurations:

ID-based Model We employ SASRec as our backbone sequential recommendation model with the following hyperparameters: maximum sequence length of 100, hidden dimension of 256, 2 attention heads, and 2 transformer blocks. For optimization, we use the Adam optimizer (Kingma and Ba, 2017) with a learning rate of 1e-4 and batch size of 128.

LLM Reranking For the reranking process, we utilize GPT-4 (Achiam et al., 2023) to refine the recommendations generated by the ID-based model and to construct the engagement profiles that capture user preferences and intentions.

Instruction Finetuning We finetune LLaMA3 8B (Touvron et al., 2023) using Low-Rank Adaptation (LoRA) (Hu et al., 2021) with a rank of 8 and alpha of 16. The finetuning process is optimized using Adam with a learning rate of 5e-4 and batch size of 32. We use LLaMA-Factory (Zheng et al., 2024b) for instruction finetuning.

Computing Resources All experiments are conducted on a single NVIDIA A100 GPU with 80GB memory. The ID-based models and all baselines are implemented in PyTorch (Paszke et al., 2019).

C Baselines

We adopt the following representative sequential recommendation models as baselines:

1. **SASRec:** (Kang and McAuley, 2018) A causal sequential model using a unidirectional transformer to predict the next item in a sequence of IDs.

2. **BERT4Rec:** (Sun et al., 2019) A sequential method using a bidirectional transformer to learn user behavior sequences for recommendations.

3. **S^3-Rec:** (Zhou et al., 2020) A self-supervised learning approach, primarily leveraging Mutual Information Maximization (MIM) during its pre-training phase to learn from the correlations between different elements.

4. **Caser:** (Tang and Wang, 2018) A sequential method treats recent item sequence embeddings as an 'image,' using horizontal convolutional filters to capture union-level patterns..

5. **TALLRec:** (Bao et al., 2023) A framework that finetunes LLMs for recommendation tasks, aligning pre-trained models with recommendation.

6. **P5:** (Geng et al., 2023) A training framework for T5, extended to LLMs. It uses personalized prompting and template-based training to unify multiple recommendation tasks.

7. **LLMRank:** (Hou et al., 2024) An LLM-based recommendation system that pairs with sequential models with careful prompting.

8. **RecLM:** (Lu et al., 2024) A framework combining supervised and reinforcement learning to improve LLMs' instruction-following abilities and generalize across various recommendation tasks.

9. **GPT-4o:** (Achiam et al., 2023) A state-of-the-art general purpose large language model from OpenAI.

D Detailed Categorization of Preferences and Intentions

User engagement profiling serves as a crucial step in understanding personalized recommendation scenarios. By examining a user's historical interactions alongside their expressed or inferred preferences, we can more accurately capture their long-term interests and current intentions. In this appendix, we provide a detailed categorization of user preferences and intentions, expanding on the definitions presented in the main text. This additional information aims to clarify how these categories can be applied to construct more nuanced user engagement profiles, ultimately leading to more effective and explainable recommendation outcomes.

Preference (P) describes a user's personalized likes or dislikes for certain product attributes or features. Preferences capture inherent, long-term interests and needs. Depending on the level of personalization, user preferences can be categorized as follows:

- *No Preference (P0)*: When the system lacks any information about the user's preferences, recommendations are non-personalized. This often occurs in cold-start situations where the system has no historical data to base recommendations on.

- *General Preferences (PC)*: Reflect interests through both direct expressions and inferred patterns expressed by the user. This includes straightforward preferences expressed through ratings and reviews, providing direct feedback. It also includes patterns observed from long-term interactions, such as browsing history and purchase activities, which reveal underlying interests. Together, these aspects form a comprehensive view of the user's personalized likes and dislikes.

- *Novelty Preferences (PN)*: Reflect the user's interest in exploring both new and unexpected content beyond their typical preferences. This includes a willingness to actively try categories or domains different from their usual choices. It also reflects an openness to items that pleasantly surprise them, even if these items do not match their established tastes. These elements together add diversity and exploration to recommendations.

Intention (I) describes a user's immediate needs and goals at a specific point in time. Unlike long-term preferences, intentions focus on the user's current, specific demands, which may differ from their usual interests. Intentions can be categorized based on their level of clarity:

- *No Intention (I0)*: The user has no clear needs, showing exploratory behavior to discover potential interests through the system's recommendations.

- *General Intention (IC)*: Reflect the user's expressed need, which can range from vague to specific. This intention can be vague, where the user describes a general goal or purpose without identifying specific product types, attributes, or features. Such expressions often lack clear guidance, requiring further refinement or exploration. Alternatively, the intention can be specific, where the user provides detailed information, explicitly outlining the characteristics, attributes, or requirements they are seeking.

- *Exploratory Intention (IE)*:Reflect the user's desire to explore and engage with new domains or product types. This intention demonstrates a purposeful approach where the user actively searches for opportunities to broaden their knowledge, experience diverse options, or discover innovative solutions that expand their understanding and satisfaction. It highlights a proactive and goal-oriented behavior in their exploration process.

E Latency Evaluation and Scalability Considerations

To assess the scalability of our approach, we conducted latency experiments using **two NVIDIA A100 GPUs** with **vLLM deployment** (Kwon et al., 2023) under simple setup configurations. Results are summarized in Table 8.

While these results indicate limitations for real-time applications in an academic setup, we believe performance can be significantly improved through industrial-grade infrastructure and optimization. Potential strategies include:

- **Infrastructure Scaling**: Deploying across more GPUs with proper load balancing.
- **Model Optimization**: Quantization, distillation, or using smaller LLM variants.
- **Caching Strategies**: Implementing result caching for frequent queries.
- **Batch Processing**: Leveraging more aggressive batch inference for improved throughput.

We acknowledge scalability challenges and discuss them in the limitation section. However, successful industrial deployments of LLM-based recommender systems demonstrate that, with proper infrastructure and optimization, these issues can be effectively addressed.

F Detailed Implementation and Results of A/B Test

Participants. We recruited 53 random participants for this study.

Procedure. Participants were presented with user profiles (including browsing history) and corresponding recommendations generated by each of the three

Metric	Value (seconds)
Average Latency	2.13
P50 Latency	2.68
P90 Latency	2.74
P95 Latency	2.77
P99 Latency	2.98
QPS (Queries per Second)	10.48

Table 8: Latency performance metrics using vLLM on two A100 GPUs.

methods: A1, A2, and B. To avoid bias, the source of the recommendation (A1, A2, or B) was not revealed to the participants. Each participant was shown the user profile and recommendations, and asked to rate the recommendations based on two metrics: relevance and serendipity. The concept of "relevance" and "serendipity" were clearly explained to the participants before the test. We employed a 5-point Likert scale for collecting the ratings. To minimize potential priming effects where explicitly considering relevance might influence the perception of serendipity, the relevance question was presented before the serendipity question. Participants were not shown the serendipity question until after completing the relevance assessment.

Groups. The control group (A) included two subgroups: recommendations from a baseline algorithm (A1) and random recommendations (A2). The experimental group (B) received recommendations generated by our LLM-based model.

- **A1**: Baseline recommendation algorithm: SASRec (Kang and McAuley, 2018).
- **A2**: Random recommendations (sample randomly according to popularity)
- **B**: SOLAR

Evaluation Metrics.

Relevance (Positive). Participants rated the relevance of each recommendation by answering: "Based on the user's browsing history, how relevant is this recommendation to their interests?" (1: Not at all relevant, 5: Highly relevant).

Serendipity (Positive). Participants rated their agreement with the statement: "This recommendation is surprising and delightful" (1: Strongly Disagree, 5: Strongly Agree).

Data Analysis. We employed the Mann-Whitney U test to compare the serendipity and relevance scores between groups (A1 vs. B and A2 vs. B). We used Spearman's rank correlation coefficient to assess the relationship between serendipity and relevance scores within each group (A1, A2, and B). A significance level of $p < 0.05$ was used.

Experimental Results.

Rating Data. Table 9 presents the raw rating data collected from the participants.

Mann-Whitney U Test and Spearman Correlation Coefficients Results. Table 10 presents the results of the Mann-Whitney U test and the Spearman correlation coefficients, respectively.

Discussion. The results demonstrate that our method (Group B) achieved significantly higher serendipity scores compared to both the baseline algorithm (A1) and random recommendations (A2) (Mann-Whitney U test, $p < 0.01$). Group B also scored significantly higher on relevance compared to both A1 and A2 (Mann-Whitney U test, $p < 0.05$). Importantly, within Group B, we observed a statistically significant positive correlation between serendipity and relevance scores (Spearman's rho = 0.28, $p < 0.05$). This suggests that the LLM is capable of generating recommendations that are both surprising and relevant to users' interests, even when relevance is assessed prior to serendipity.

Limitations. The offline nature of this A/B test has inherent limitations. The sample size of 53 participants, while sufficient for initial validation, may not fully represent the broader user population. Additionally, subjective ratings may be influenced by individual biases. Future work should involve a larger-scale online A/B test to further validate these findings in a real-world setting.

G Construction of SUN and SUN^{-1}

In this appendix, the two figures (Figure 6 and Figure 7), we illustrate two examples of the overall process framework that transforms user interaction records and system instructions into recommendation outputs. In the first part (yellow), the system will receive instructions as input. In the second part (red), the user engagement profile serves as the foundation, combining the user's history with candidate items and dynamically selecting the most relevant next recommendation through various methods at output (green), including generative, direct recommendation, reranking, and matching. As the reverse method, the user history and potential recommendation results serve as input (red) to prompt the model to output the user engagement profile (green). And detailed templates are presented in Table 13, Table 14 and Table 15.

H Templates of Generation of Engagement Profile

In this appendix, we present detailed templates for generating engagement profiles based on multiple datasets, including Movielens, Booksand Movies & TV, see Table 16,Table 17, and Table 18. These templates leverage users' historical interaction data (and corresponding reviews) to extract and infer various dimensions of user preferences and intentions. In these templates, {interaction} and {reviews} serve as placeholders for user interaction histories and associated feedback, while {constraint} introduces necessary limiting conditions.Using these templates, the system improves recommendation

Table 9: Raw Serendipity and Relevance Ratings

Group	Serendipity_1	Serendipity_2	Serendipity_3	Serendipity_4	Serendipity_5	Relevance_1	Relevance_2	Relevance_3	Relevance_4	Relevance_5
A1	2	16	13	14	8	3	8	15	23	4
A2	6	14	19	8	6	9	16	17	7	4
B	4	3	12	15	19	1	7	7	17	21

Table 10: Mann-Whitney U Test and Spearman Correlation Coefficients Results

Table 11: Mann-Whitney U Test

Comparison	U Value	p-value
A1 vs. B - Serendipity	980.0	0.002913257250473625
A2 vs. B - Serendipity	809.0	5.581316015540164e-05
A1 vs. B - Relevance	920.5	0.000729550390629055
A2 vs. B - Relevance	606.5	1.2171162949477135e-07

Table 12: Spearman Correlation

Group	rho	p-value
A1	0.043	0.760
A2	-0.178	0.203
B	0.283	0.040

accuracy and serendipity.

I Artifact License and Usage

We release the code used in our experiments under the MIT License.[2] All external artifacts used in this work (including datasets and tools) conform to their intended usage as specified by the original authors or sources, and are used solely for non-commercial academic research.

All released artifacts are intended for non-commercial academic use only.

[2] https://opensource.org/licenses/MIT

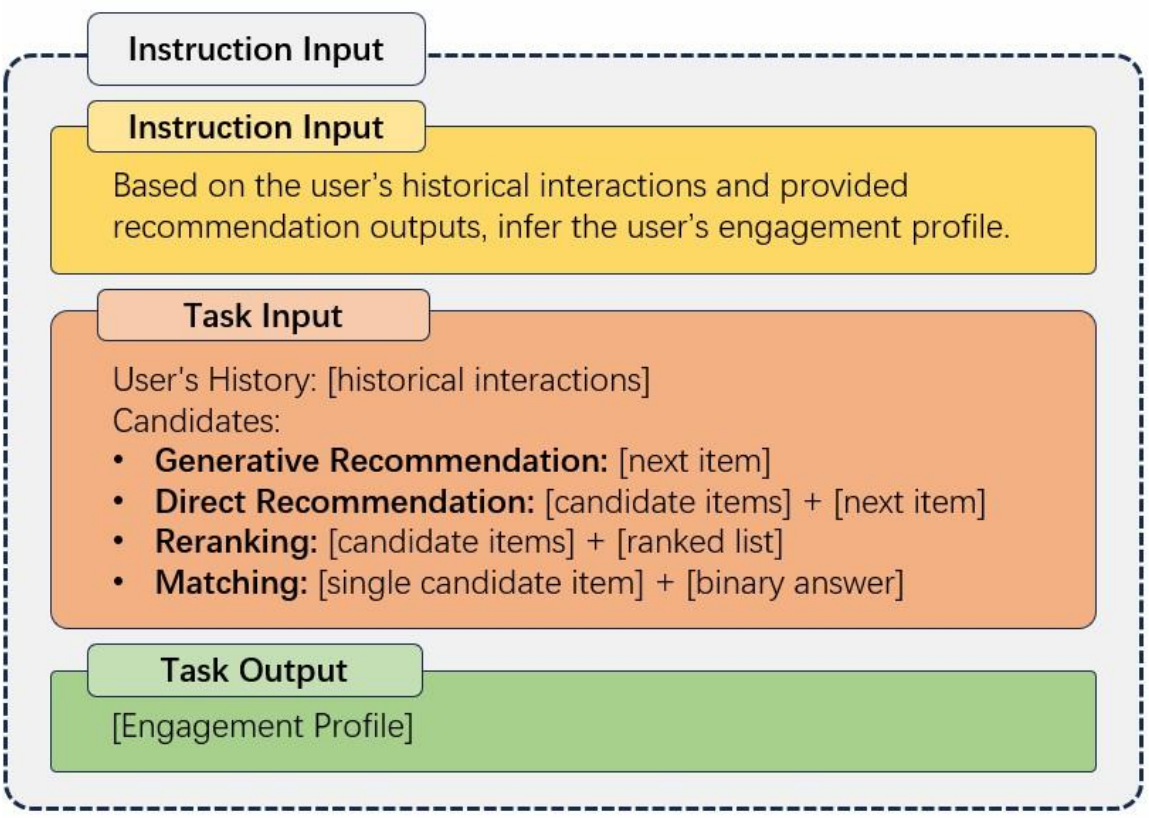

Figure 6: Reverse Recommendation Task.

	Templates of Construction of SUN (RecTempalte)
(1)	{reranking} The behavioral sequence of the user is shown below: {historical_interactions}, which can be used to infer the user's preferences {explicit_preference}. Then please rerank the items to better align with the user's preferences by comparing the candidates and their similarities to the user's preferences. The candidates are: {candidate_items}.
(2)	{reranking} You have some information about this user, which is shown below: {explicit_preference}, the user's historical interactions: {historical_interactions} Based on this information, please recommend the reranking order of items for the user, which should match the user's preference, from the following candidates: {candidate_items}
(3)	{generation} Using the user's historical interactions as input data, predict the next product that the user is most likely to interact with. The historical interactions are provided as follows: {historical_interactions}.
(4)	{generation} Given the user's interaction history: {historical_interactions}, what is the optimal product to suggest next?
(5)	{generation} Given the sequence of the user's past interactions: {historical_interactions}, what is the most suitable product to recommend next?
(6)	{generation} Considering the user's interaction pattern: {historical_interactions}, suggest the next likely product they would engage with.
(7)	{generation} Given the historical context of user interactions: {historical_interactions}, what is the optimal next product recommendation?
(8)	{generation} Based on the user's historical engagement data: {historical_interactions}, provide the next product recommendation.
(9)	{generation} Based on the user's past interaction data: {historical_interactions}, suggest the most relevant product for their next interaction.
(10)	{generation} Using the provided interaction history: {historical_interactions}, determine the most likely product the user would engage with next.
(11)	{generation} You are a recommendation system, and are good at recommending products to a user based on his preferences. Given the user's preferences: {explicit_preference}, please recommend products that are consistent with those preferences.
(12)	{generation} As we know, a user's behavior is driven by his preferences, which determine what they are likely to buy next. Your task is to predict what products a user will purchase next, based on his preferences. Given the user's preferences as follows: {explicit_preference}, please make your prediction.
(13)	{generation} Given the following historical interaction of the user: {historical_interactions}. You can infer the user's preference: {explicit_preference}. Please predict next possible item for the user.
(14)	{generation} To make a recommendation for this user, we need to analyze their historical interactions: {historical_interactions}. As we know, historical interactions reflect the user's preferences {explicit_preference}. Based on these preferences, please recommend an item that you think would be suitable for them.
(15)	{generation} Recommend the next potential product to a user based on his profile and past interactions. You have access to the user's profile information, including his preference: {explicit_preference} and past interactions: {historical_interactions}. For example, if the user recently interacted with {recent_item}, you might consider similar products. Now, based on this approach, determine what product would be recommended to him next.
(16)	{generation} Imagine the user recently interacted with {recent_item}. Using this example, and given the user's historical interactions as input data: {historical_interactions}, predict the next product that the user is most likely to interact with.
(17)	{direct} The user has previously purchased the following items: {historical_interactions}. This information indicates their personalized preferences {explicit_preference}. Based on this information, is it likely that the user will interact with {candidate_item} next?
(18)	{direct} Based on the user's historical interaction list, which is provided as follows: {historical_interactions}, you can infer the user's personalized preference {explicit_preference}. And your task is to use this information to predict whether the user will click on {candidate_item} next.
(19)	{direct} Please recommend an item to the user based on the following information about the user: {historical_interactions}, the user's historical interaction, which is as follows: {explicit_preference} Try to select one item from the following candidates that is consistent with the user's preference: {candidate_items}.
(20)	{generation} Suppose you are a search engine, now the user search that {explicit_preference_vague_intention_specific_intention}, can you generate the item to respond to user's query?

Table 13: Generation templates for the Recommendation Task

	Templates of Construction of SUN (RecTemplateInverse)
(1)	{explicit_preference} The behavioral sequence of the user is shown below: {historical_interactions}. The candidates were provided as: {candidate_items}, and they have been reranked to better align with the user's preferences: {rerank_list}. Based on this information, please infer the user's explicit preferences that likely led to this reranking.
(2)	{explicit_preference} You have observed that the user has clicked on the following items: {historical_interactions}. The following candidates were presented: {candidate_items}, and they have been reranked in an order deemed suitable for the user: {rerank_list}. Based on this information, please infer the user's explicit preferences that likely led to this reranking.
(3)	{explicit_preference} You have some information about this user, which is shown below: the user's historical interactions: {historical_interactions}. The candidates presented were: {candidate_items}, and they have been reranked in the following order: {rerank_list}. Based on this information, please infer the user's explicit preferences that would justify this reranking.
(4)	{implicit_preference} The user has interacted with the following items in the past: {historical_interactions}. The candidates provided were: {candidate_items}, and they have been reranked to better align with the user's interests: {rerank_list}. Based on this information, please infer the user's implicit preferences that likely led to this reranking.
(5)	{vague_intention} The user has shown the following historical interactions: {historical_interactions}, and the candidate items were provided as: {candidate_items}. The candidates have been reranked in this order: {rerank_list}. Based on this information, infer the user's vague intention that could explain why this reranking aligns with their preferences.
(6)	{specific_intention} Analyzing the user's past behavior: {historical_interactions} and the given candidates: {candidate_items}, which have been reordered to: {rerank_list}, please determine the user's specific intention that could explain this preference for certain elements over others.
(7)	{explicit_preference} Given the following historical interaction of the user: {historical_interactions}. And the next recommended item: {next_item}. Please infer the user's explicit preferences that would likely lead to this recommendation.
(8)	{novelty_preference} Given the user's historical behavior and intention: {historical_interactions}, and the next recommended item: {next_item}, please infer the user's exploratory preferences that would justify this recommendation.
(9)	{specific_intention} Given the following historical interactions of the user: {historical_interactions}, and the next recommended item: {next_item}. Please infer the specific intention that would likely lead to this recommendation, such as seeking a particular genre, theme, or type of item.
(10)	{specific_intention} To better understand the user's needs, consider their past interactions: {historical_interactions}. The next recommended item is: {next_item}. Based on this information, infer the user's specific intention that would justify this recommendation, focusing on concrete preferences or desires.
(11)	{exploratory_intention} The user has recently been recommended the following item: {next_item}. Given the user's historical actions: {historical_interactions} and the candidates: {candidate_items}, please infer the user's exploratory intention that would justify this surprising recommendation.
(12)	{exploratory_intention} The user was recommended the following item: {next_item}. Considering their historical interactions: {historical_interactions} and the set of candidates: {candidate_items}, please infer the user's lack of specific intention for surprising recommendations that justify the selection of this item.
(13)	{explicit_preference} Please try to infer the preference to the user based on the following information: {historical_interactions}, the user's historical interaction, which is as follows: {next_item} and the candidate item: {candidate_items}.
(14)	{vague_intention} The user has received the following recommendation: {next_item}. Given their historical actions: {historical_interactions} and the set of candidates: {candidate_items}, please infer the user's vague intention that could justify this recommendation.
(15)	{implicit_preference} Based on the user's historical interaction list: {historical_interactions}, and considering the candidate items: {candidate_items}, the item most likely to be clicked next is: {next_item}. Please infer the user's implicit preferences that would justify the selection of this item.

Table 14: Generation templates for the Reverse Recommendation Task

	Templates of Construction of SUN (RecTemplateSeren)
(1)	{generation} The user likes to explore new types of products and has recently shown interest in items that differ from their usual preferences. The user is looking to try new domains or product types. Based on the user's historical behavior and intention: {historical_interactions}, generate a product recommendation that aligns with the user's novelty preference: {novelty_preference}.
(2)	{generation} The user is interested in exploring new types of products while maintaining certain explicit preferences: {explicit_preference}. Given the user's exploratory intention ({exploratory_intention}) to try something new and different, please generate a product recommendation that aligns with both the user's explicit preferences and their desire for exploration.
(3)	{direct} The user enjoys receiving surprising recommendations and wants to try items that do not match their usual preferences. Based on the user's exploratory intention:{exploratory_intention} and combine the user's historical action : {historical_interactions}, select the item most likely to offer a pleasant surprise from the following candidates: {candidate_items}
(4)	{matching} The user is interested in new types of products that do not match their usual preferences:{explicit_preference} but their needs are still unclear. Please determine whether the following item matches the user's vague exploratory intention and answer "Yes" or "No": {candidate_item}
(5)	{direct} The user has no specific intention but enjoys receiving surprising recommendations. Based on this, select the item most likely to provide a pleasant surprise from the following candidates: {candidate_items}
(6)	{matching} The user enjoys being surprised and has shown implicit preferences based on their historical interactions: {historical_interactions}. The user's current intention may be vague as following : {vague_intention}. Based on this information, evaluate the following candidate item: {candidate_item} to determine if it would be a suitable recommendation for the user, please answer "Yes" or "No" for the fitness of candidate.
(7)	{generation} You are a search engine. Here is the historical interaction of a user: {historical_interactions}. And his personalized preferences are as follows: {explicit_preference}. Your task is to generate a new product that are consistent with the user's preference.
(8)	{generation} The user has interacted with a list of items, which are as follows: {historical_interactions}. Based on these interacted items, the user current intent are as follows {vague_intention}, and your task is to generate products that match the user's current intent.
(9)	{generation} As a search engine, you are assisting a user who is searching for the query: {specific_intention}. Your task is to recommend products that match the user's query and also align with their preferences based on their historical interactions, which are reflected in the following: {historical_interactions}
(10)	{direct} Using the user's current query: {explicit_preference_vague_intention_specific_intention} and their historical interactions: {historical_interactions} you can estimate the user's preferences {explicit_preference}. Please respond to the user's query by selecting an item from the following candidates that best matches their preference and query: {candidate_items}
(11)	{direct} The user wants to try some products and searches for: {explicit_preference_vague_intention_specific_intention}. In addition, they have previously bought: {historical_interactions}. You can estimate their preference by analyzing his historical interactions. {explicit_preference} Please recommend one of the candidate items below that best matches their search query and preferences: {candidate_items}

Table 15: Generation templates for the RecTemplate for Serendipity Purpose

	Templates of Generation of Engagement Profile (Movielens)
(1)	User's historical interactions: {interaction}. Based on these movie titles, use your knowledge to generate a description of the user's implicit preferences, such as their favorite genres, themes, or notable patterns. {constraint}
(2)	The user has browsed the following movies in chronological order: {interaction}. Based on this browsing history, use your understanding of these movies to generate a description of the user's implicit preferences, including their likely favorite genres, themes, or types of movies. {constraint}
(3)	Recently, the user has browsed the following movies: {interaction}. Based on this recent activity, apply your knowledge of these movie to generate a description of the user's current movie preferences, focusing on genres, themes, or other noticeable patterns.{constraint}
(4)	Analyze the user's recent viewing history: {interaction}. From these interactions, use your knowledge of these movies to infer the user's implicit preferences, such as preferred genres, sub-genres, or specific types of storylines. {constraint}
(5)	The user has shown a strong interest in the following movies: {interaction}. Using this data, infer their explicit preferences, such as particular themes, moods, or types of narratives they actively seek.{constraint}
(6)	Consider the user's engagement with the following movies: {interaction}. Based on these patterns, determine their explicit preferences, such as favorite directors, frequent actors, or recurring motifs that they seem to appreciate. {constraint}
(7)	The user has recently browsed a variety of different movie genres: {interaction}. Based on this diverse viewing pattern, describe the user's novelty preferences, such as their openness to exploring new genres or trying unexpected movie types. {constraint}
(8)	Given the user's browsing history: {interaction}, identify any novelty preferences they may have, such as a willingness to explore genres outside their usual interest or a desire for unique and unconventional film experiences. {constraint}
(9)	The user has moved from browsing typical genres to less common ones: {interaction}. Describe the user's novelty preferences, focusing on their interest in discovering diverse genres or unique cinematic styles. {constraint}
(10)	The user has recently watched the following movies: {interaction}. Reflect on this history to infer a general type or mood of movies they might be interested in next, without narrowing down to a specific genre or characteristic. {constraint}
(11)	Given the user's recent viewing history: {interaction}, suggest a broad intention for what they may want to watch next, focusing on an overall style or feeling rather than pinpointing a particular movie or specific genre. {constraint}
(12)	Based on these movies: {interaction}, generate an open-ended intention that represents a general mood or broad category the user could be leaning towards, even if their specific preferences aren't clear. {constraint}
(13)	The user has watched these movies: {interaction}. Use this data to determine a specific movie intention they might have, such as seeking a particular genre, a specific plot, or a film with certain defining characteristics. {constraint}
(14)	Based on the user's recent movie list: {interaction}, infer a clearly defined intention for the next type of film they may want to watch, focusing on particular elements like genre, theme, or distinctive features. {constraint}
(15)	Considering the user's viewing pattern: {interaction}, determine a specific intention about the next movie they are likely to watch, including precise details about the genre, mood, or main elements they are interested in. {constraint}
(16)	The user has recently watched these movies:{interaction}. Based on this history, suggest an exploratory intention where the user might want to explore genres or types of movies they haven't typically watched. {constraint}
(17)	Given the user's movie-watching history of: {interaction}, infer an exploratory intention indicating their curiosity to explore new and different genres, styles, or narrative types that they might not have considered before. {constraint}
(18)	Using the following interaction data: {interaction}, generate an exploratory intention for the user, where they express interest in trying out new genres, themes, or movie types that differ from their usual choices. {constraint}

Table 16: Generation templates for the Movielens dataset

	Templates of Generation of Engagement Profile (Books)
(1)	Analyze the user's reading history: {interaction} and the associated reviews: {reviews}. From these data points, determine the user's explicit preferences, such as the genres, themes, or specific book characteristics they explicitly praise or mention in their comments. {constraint}
(2)	Considering the user's reading history of these books: {interaction}, along with their corresponding reviews: {reviews}, generate a description of the user's explicit preferences, focusing on any recurring genres, themes, or patterns evident in their comments. {constraint}
(3)	Based on the user's recent engagement with the following books: {interaction} and their comments: {reviews}, identify their explicit preferences by analyzing the sentiments and focus of their reviews, such as preferred genres, themes, or author styles they frequently mention or praise. {constraint}
(4)	Analyze the user's reading history: {interaction} and the associated reviews: {reviews}. From these data points, determine the user's explicit preferences, such as the genres, themes, or specific book characteristics they explicitly praise or mention in their comments. {constraint}
(5)	The user has shown a clear interest in certain books: {interaction}, with specific comments: {reviews}. Using this data, infer their explicit preferences, such as favorite themes, plot types, or narrative styles they often highlight in their reviews. {constraint}
(6)	Consider the user's engagement with these books: {interaction}, accompanied by their reviews: {reviews}. Based on these reviews, identify explicit preferences, such as preferred authors, frequent genres, or writing styles that the user frequently praises or critiques. {constraint}
(7)	The user has recently reviewed a variety of different genres or unconventional books: {interaction}, with comments: {reviews}. Describe the user's novelty preferences, such as their openness to experimenting with new genres or exploring unique literary styles, based on the diversity of their reviews. {constraint}
(8)	Given the user's diverse reading history: {interaction} and their reviews: {reviews}, identify any novelty preferences they may have, such as a tendency to seek out unique literary experiences or genres that are outside their usual interests. {constraint}
(9)	The user has moved from reading typical genres to exploring less common ones: {interaction}, as indicated by their reviews: {reviews}. Describe the user's novelty preferences, focusing on their interest in discovering new genres or unconventional narrative approaches. {constraint}
(10)	The user has recently read the following books: {interaction}, with the following reviews: {reviews}. Reflect on this history and the accompanying reviews to infer a general type or mood of books they might be interested in next, without narrowing down to a specific genre or characteristic. {constraint}
(11)	Given the user's recent reading history: {interaction} and their reviews: {reviews}, suggest a broad intention for what they may want to read next, focusing on an overall style or feeling rather than pinpointing a particular book or specific genre. {constraint}
(12)	Based on these books: {interaction} and the corresponding reviews: {reviews}, generate an open-ended intention that represents a general mood or broad category the user could be leaning towards, even if their specific preferences aren't clear. {constraint}
(13)	The user has read these books: {interaction} and provided the following reviews: {reviews}. Use this data to determine a specific book intention they might have, such as seeking a particular genre, a specific plot, or a book with certain defining characteristics. {constraint}
(14)	Based on the user's recent book list: {interaction} and their reviews: {reviews}, infer a clearly defined intention for the next type of book they may want to read, focusing on particular elements like genre, theme, or distinctive features. {constraint}
(15)	Considering the user's reading pattern: {interaction} and their reviews: {reviews}, determine a specific intention about the next book they are likely to read, including precise details about the genre, mood, or main elements they are interested in. {constraint}
(16)	The user has recently read these books: {interaction} and left the following reviews: {reviews}. Based on this history, suggest an exploratory intention where the user might want to explore genres or types of books they haven't typically read. {constraint}
(17)	Given the user's book-reading history of: {interaction} and their reviews: {reviews}, infer an exploratory intention indicating their curiosity to explore new and different genres, styles, or narrative types that they might not have considered before. {constraint}
(18)	Using the following interaction data: {interaction} and corresponding reviews: {reviews}, generate an exploratory intention for the user, where they express interest in trying out new genres, themes, or book types that differ from their usual choices. {constraint}

Table 17: Generation templates for the Books dataset

	Templates of Generation of Engagement Profile (Movies & TV)
(1)	Analyze the user's viewing history: {interaction} and the associated reviews: {reviews}. From these data points, determine the user's explicit preferences, such as the genres, themes, or specific movie characteristics they explicitly praise or mention in their comments. {constraint}
(2)	Considering the user's viewing history of these movies/TV shows: {interaction}, along with their corresponding reviews: {reviews}, generate a description of the user's implicit preferences, focusing on any recurring genres, themes, or patterns evident in their comments. {constraint}
(3)	Based on the user's recent engagement with the following movies/TV shows: {interaction} and their comments: {reviews}, identify their implicit preferences by analyzing the sentiments and focus of their reviews, such as preferred genres, themes, or character types. {constraint}
(4)	Analyze the user's viewing history: {interaction} and the associated reviews: {reviews}. From these data points, determine the user's explicit preferences, such as the genres, themes, or specific movie characteristics they explicitly praise or mention in their comments. {constraint}
(5)	The user has shown a clear interest in certain movies/TV shows: {interaction}, with specific comments: {reviews}. Using this data, infer their explicit preferences, such as favorite themes, plot types, or emotional tones they often highlight in their reviews. {constraint}
(6)	Consider the user's engagement with these movies/TV shows: {interaction}, accompanied by their reviews: {reviews}. Based on these reviews, identify explicit preferences, such as preferred directors, frequent actors, or narrative styles that the user frequently praises or critiques. {constraint}
(7)	The user has recently reviewed a variety of different genres or unconventional movies/TV shows: {interaction}, with comments: {reviews}. Describe the user's novelty preferences, such as their openness to experimenting with new genres or exploring unique cinematic styles, based on the diversity of their reviews. {constraint}
(8)	Given the user's diverse viewing history: {interaction} and their reviews: {reviews}, identify any novelty preferences they may have, such as a tendency to seek out unique film experiences or genres that are outside their usual interests. {constraint}
(9)	The user has moved from watching typical genres to exploring less common ones: {interaction}, as indicated by their reviews: {reviews}. Describe the user's novelty preferences, focusing on their interest in discovering new genres or unconventional storytelling approaches. {constraint}
(10)	The user has recently watched the following movies: {interaction}, with the following reviews: {reviews}. Reflect on this history and the accompanying reviews to infer a general type or mood of movies they might be interested in next, without narrowing down to a specific genre or characteristic. {constraint}
(11)	Given the user's recent viewing history: {interaction} and their reviews: {reviews}, suggest a broad intention for what they may want to watch next, focusing on an overall style or feeling rather than pinpointing a particular movie or specific genre. {constraint}
(12)	Based on these movies: {interaction} and the corresponding reviews: {reviews}, generate an open-ended intention that represents a general mood or broad category the user could be leaning towards, even if their specific preferences aren't clear. {constraint}
(13)	The user has watched these movies: {interaction} and provided the following reviews: {reviews}. Use this data to determine a specific movie intention they might have, such as seeking a particular genre, a specific plot, or a film with certain defining characteristics. {constraint}
(14)	Based on the user's recent movie list: {interaction} and their reviews: {reviews}, infer a clearly defined intention for the next type of film they may want to watch, focusing on particular elements like genre, theme, or distinctive features. {constraint}
(15)	Considering the user's viewing pattern: {interaction} and their reviews: {reviews}, determine a specific intention about the next movie they are likely to watch, including precise details about the genre, mood, or main elements they are interested in. {constraint}
(16)	The user has recently watched these movies: {interaction} and left the following reviews: {reviews}. Based on this history, suggest an exploratory intention where the user might want to explore genres or types of movies they haven't typically watched. {constraint}
(17)	Given the user's movie-watching history of: {interaction} and their reviews: {reviews}, infer an exploratory intention indicating their curiosity to explore new and different genres, styles, or narrative types that they might not have considered before. {constraint}
(18)	Using the following interaction data: {interaction} and corresponding reviews: {reviews}, generate an exploratory intention for the user, where they express interest in trying out new genres, themes, or movie types that differ from their usual choices. {constraint}

Table 18: Generation templates for the Movies & TV datasets

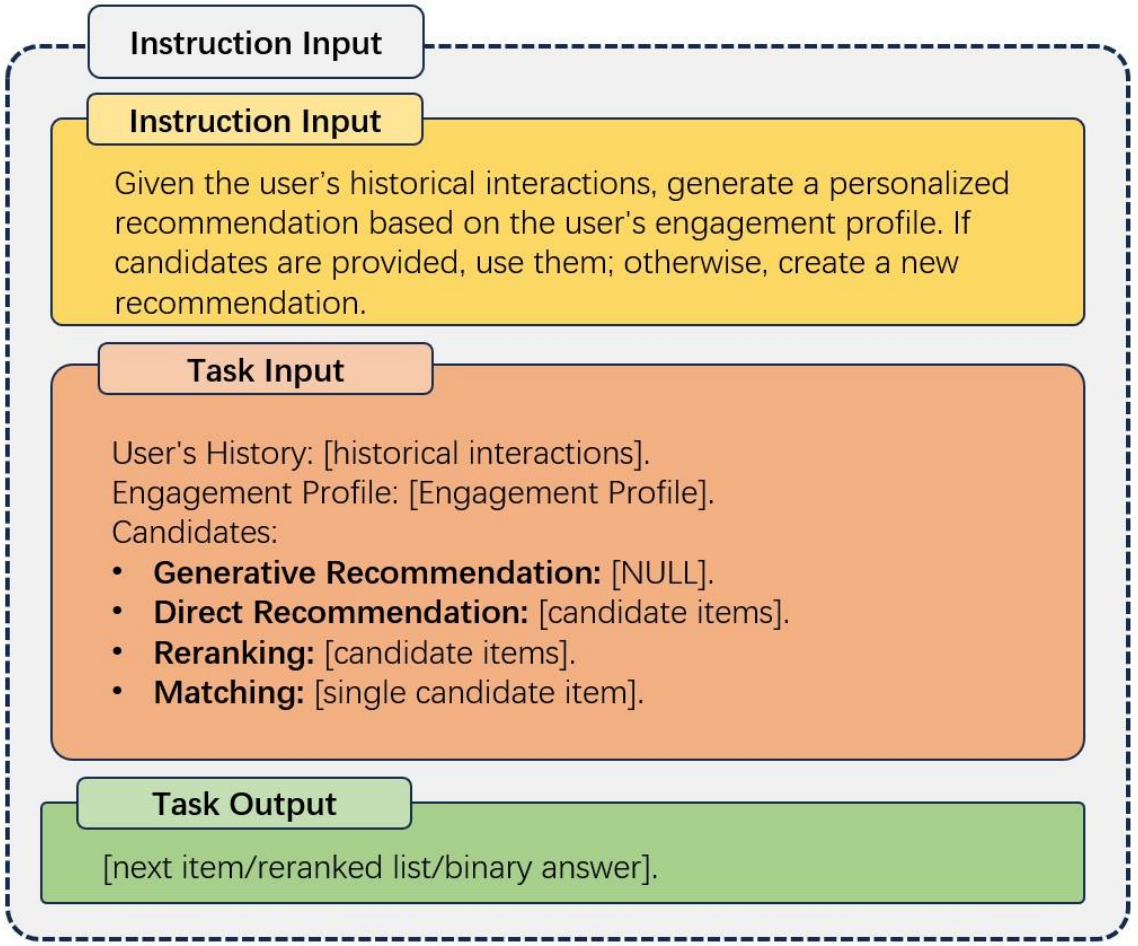

Figure 7: Recommendation Task.