

Large Language Models for Intent-Driven Session Recommendations

Zhu Sun^{1,2,*}, Hongyang Liu^{3,*}, Xinghua Qu^{4,†}, Kaidong Feng⁵, Yan Wang³, Yew-Soo Ong^{1,6}

¹Center for Frontier AI Research, A*STAR, Singapore

² Institute of High Performance Computing, A*STAR, Singapore

³Macquarie University, Australia

⁴Shanda Group AI Lab, Singapore

⁵Yan Shan University, China

⁶Nanyang Technological University, Singapore

* denotes co-first authors; † denotes corresponding author: teddy.qu@shanda.com

ABSTRACT

Intent-aware session recommendation (ISR) is pivotal in discerning user intents within sessions for precise predictions. Traditional approaches, however, face limitations due to their presumption of a uniform number of intents across all sessions. This assumption overlooks the dynamic nature of user sessions, where the number and type of intentions can significantly vary. In addition, these methods typically operate in latent spaces, thus hinder the model's transparency. Addressing these challenges, we introduce a novel ISR approach, utilizing the advanced reasoning capabilities of large language models (LLMs). First, this approach begins by generating an initial prompt that guides LLMs to predict the next item in a session, based on the varied intents manifested in user sessions. Then, to refine this process, we introduce an innovative prompt optimization mechanism that iteratively self-reflects and adjusts prompts. Furthermore, our prompt selection module, built upon the LLMs' broad adaptability, swiftly selects the most optimized prompts across diverse domains. This new paradigm empowers LLMs to discern diverse user intents at a semantic level, leading to more accurate and interpretable session recommendations. Our extensive experiments on three real-world datasets demonstrate the effectiveness of our method, marking a significant advancement in ISR systems.

CCS CONCEPTS

- Information systems → Recommender systems;
- Computing methodologies → Neural networks.

KEYWORDS

Session-based Recommendation, User Intents, Large Language Models, Prompt Optimization

1 INTRODUCTION

Session-based recommendation (SR) [5, 14, 20, 25, 57] aims to predict the next interacted item based on short anonymous behavior sessions. Typically, different sessions may unveil diverse user intents [50]. Figure 1 illustrates two real sessions in Amazon Electronic dataset [34], where the first session in (a) reflects one main purpose, i.e., shopping for laptop and accessories, whereas the second one in (b) is associated with two major intents, i.e., shopping for laptop protectors and camera accessories, respectively. However, most public datasets do not include explicit intents of a session as



(a) Session 1 - Laptop and accessories



(b) Session 2 - Laptop protectors in green border and camera accessories in pink border

Figure 1: Examples of user sessions with various intents.

it may be intrusive and disruptive to ask users about their current session purposes directly [35]. Hence, intent-aware session recommendation (ISR) has emerged to capture the latent user intents within a session, thus enhancing the accuracy of SR.

Specifically, early studies in ISR [26, 32, 51] primarily constrain sessions to a single purpose or goal, such as shopping for a laptop and accessories in Figure 1(a). However, this simplistic assumption doesn't always hold in real-world scenarios, where a session may involve diverse items for various purposes, as illustrated in Figure 1(b). It thus poses barriers for such methods seeking further performance improvements. Consequently, various approaches have been developed to model multiple intentions within a session, e.g., IDSR [9], MCPRN [50], and NirGNN [23]. Despite the success, they suffer from two major limitations. First, they rely on an unrealistic assumption that all sessions possess a consistent and fixed number of intentions, treating this as a hyper-parameter. Second, they are limited to learning latent intentions solely within the embedding space, greatly impeding the transparency of ISR. Such limitations thus further hinder these approaches from delivering more accurate and comprehensible recommendations.

Fortunately, the rise of large language models (LLMs) has opened up unprecedented opportunities in the field of ISR. LLMs, armed with advanced reasoning capabilities, have found widespread application in general recommendation scenario [4, 11, 16, 43], but have been relatively underexplored in the context of SR. In the limited body of research on SR, LLMs are generally employed in two distinct ways, namely in-context learning (ICL) [21, 49] and parameter-efficient fine-tuning [3, 64]. However, LLMs cannot fully realize their potential through simple ICL (e.g., zero-shot prompting [49]). While fine-tuning LLMs holds promise, it grapples with challenges stemming from computational demands and the availability of open-source LLMs.

Therefore, we propose a simple yet effective paradigm to exploit the power of LLMs for more effective ISR from the perspective of prompt optimization (abbreviated as PO4ISR). It is equipped with Prompt Initialization (*PromptInit*), Prompt Optimization (*PromptOpt*), and Prompt Selection (*PromptSel*). In particular, *PromptInit* aims to create an initial prompt that guides LLMs to dynamically understand session-level user intents, and predict the next item accordingly. Inspired by the study on automatic prompt optimization [38] in natural language processing (NLP), *PromptOpt* seeks to automatically optimize the initial prompt with self-reflection. To be specific, the LLM is required to offer reasoning rooted in the identified errors to improve (refine and augment) the initial prompt. The performance of improved prompts is then assessed with UCB bandits [1], thereby helping shortlist promising prompt candidates for an iterative optimization. Lastly, *PromptSel* prioritizes the selection of optimized prompts by utilizing the robust generalizability of LLMs across diverse domains, aiming to maximize accuracy improvements. As such, PO4ISR can efficiently direct LLMs to infer and comprehend dynamic user intents at a semantic level, resulting in more accurate and understandable SR.

Contributions. Our main contributions lie three-fold. (1) We introduce a simple yet powerful paradigm – PO4ISR – to utilize the capabilities of LLMs for enhanced ISR through prompt optimization. (2) The PO4ISR paradigm, composed of prompt initialization, optimization, and selection modules, empowers LLMs to semantically comprehend varying user intents in a session, resulting in more accurate and comprehensible SR. (3) Experiments on real-world datasets demonstrate that PO4ISR significantly outperforms baselines with an average improvement of 57.37% and 61.03% on HR and NDCG, respectively. Meanwhile, several insightful observations are gained, for example, (a) PO4ISR yields promising accuracy with only a small number of training samples; (b) PO4ISR exhibits advanced generalizability and excels in cross-domain scenarios; (c) PO4ISR showcases superior strength on sparser datasets with shorter sessions compared; however, it might exhibit increased hallucination tendencies with sparser datasets; (d) the performance of PO4ISR shows a positive correlation with the quality of the initial prompt, and lower-quality initial prompts tend to yield more significant improvements; and (f) a streamlined description and subtask division can enhance the quality of initial prompts.

2 RELATED WORK

We first provide an overview of the development of session-based recommendation (SR). Then, intent-aware session recommendation (ISR) is introduced, followed by the LLM-based SR.

2.1 Session-based Recommendation

Early works employ *conventional methods*, such as frequent sequential patterns [2, 60], session-level item-similarity (e.g., SKNN [22, 33]), Markov chain (e.g., FPMC [42]) and random walk [10]. Later, *recurrent neural networks* (RNN) have been applied to handle longer sequences assuming that adjacent items in a session are sequentially dependent. GRU4Rec [18] is the representative model, which has been further extended using, e.g., data augmentation [48, 52], new losses [17], parallel architecture with item features [19], and cross-session information transfer [41]. Subsequently, the *attention*

mechanism has been introduced in SR to relax this assumption by emphasizing more informative items in sessions [44], such as NARM [26] and STAMP [32]¹. To better model the high-order transition among items, *graph neural network* (GNN) based methods have been recently designed to generate more accurate item embeddings from the session graph, such as SR-GNN [54], FGNN [40], GC-SAN [56], GSL4Rec [53], GNG-ODE [13], KMVG [6], and ADRL [7]. Besides, many studies propose to better leverage both the intra- and inter-session information, e.g., GCE-GNN [51], HG-GNN [36], DGNN [12], SPARE [37], CGSR [62], and HADCG [45]. Meanwhile, other works (e.g., S²-DHCN [55] and CoHHN [66]) employ hypergraphs to enhance item representations for SR.

2.2 Intent-aware Session Recommendation

An essential line of research learns the intents hidden in the session for accurate SR. Early studies assume items inside a session are associated with one (implicit) purpose. In particular, NARM [26] extracts the last hidden state as the session representation and applies an attention mechanism on all hidden states to learn the user’s intention. STAMP [32] explicitly emphasizes the current interest reflected by the last click to capture the hybrid features of current and general interests from previous clicks via an attention network. SR-GNN [54], GC-SAN [56], GCE-GNN [51] and TAGNN [63] model each session as graph-structured data and apply GNN to learn the representations of user intents with attention networks. LESSR [8] learns local and global interests to represent user intent by easing the information loss issue. MSGAT [39] uses similar sessions to efficiently generate the session intent representation. Nonetheless, a session may encompass items with varying intentions. Thus, solely modeling the main intent could lead to information loss, potentially hurting the performance of SR.

Consequently, recent studies endeavor to learn multiple intents for more effective SR. Specifically, NirGNN [23] learns dual intents by exploiting attention mechanisms and data distribution in the session graph. MCPRN [50] designs mixture-channel purpose routing networks to detect the purposes of each item in a session and assign them to the corresponding channels. IDSR [9] projects the item representation into multiple spaces indicating various intentions and employs self-attention within each space to capture distinct intentions. HIDE [29] splits an item embedding into multiple chunks to represent various intentions and then organizes items in a session with hyperedges to help learn the associated intents. MIHSG [12] and Atten-Mixer [65] learn multi-granularity consecutive user intents to generate more accurate session representations. STAGE [28] and ISCON [35] capture the impact of multiple intrinsic intents for better SR. DAGNN [58] extracts session demands over the item category space to capture semantically correlated categories. However, they make an impractical assumption that all sessions have a uniform and fixed number of intentions. Moreover, most of them can only learn latent intents, thus restricting the transparency of SR. In contrast, we aim to leverage the advanced reasoning capabilities of LLMs to uncover varying numbers of semantic intents within a session for more accurate and comprehensible SR.

¹Our study focuses on session-based recommendation. Therefore, some popular sequential-based recommendation approaches, e.g., Bert4Rec [46] and SASRec [24] are out of our scope. Please refer to [61] for the detailed difference between session- and sequential-based recommendation.

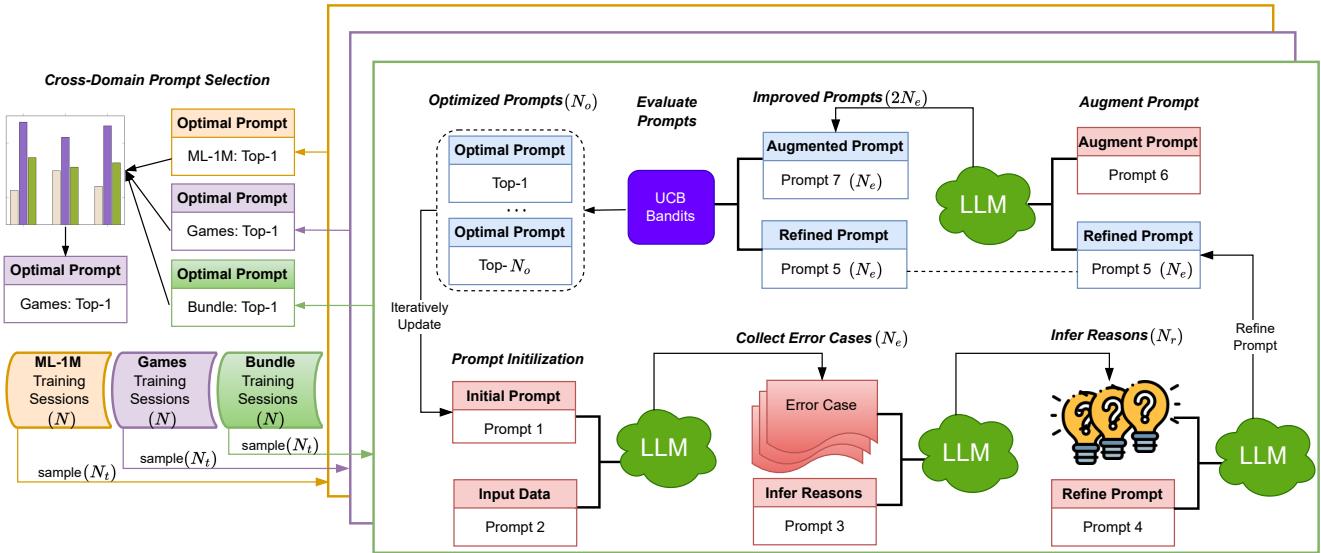


Figure 2: The overall architecture of our proposed PO4ISR paradigm.

2.3 LLM-based Session Recommendation

LLMs have achieved remarkable achievements for general recommendation [4, 11, 16, 43]. To the best of our knowledge, NIR [49] is the only one that adopts zero-shot prompting for SR, and most methods target sequential recommendation [15]. Among them, some leverage the in-context learning (ICL) capability of LLMs, for instance, Hou et al. [21] use LLMs as rankers by designing sequential, recency-based and ICL prompting. Others align LLMs for recommendation via parameter-efficient fine-tuning. To be specific, BIGRec [3] employ LLMs in an all-rank scenario by grounding LLMs to the recommendation and actual item spaces. TransRec [30] identifies fundamental steps of LLM-based recommendation to bridge the item and language spaces. GPT4Rec [27] generates multiple queries given item titles in a user’s history with beam search and then retrieves items for recommendation by searching these queries with a search engine. LlamaRec [64] uses an ID-based sequential recommender as the retriever to generate candidates and then designs a verbalizer to transform LLM output as the probability distribution for ranking. Moreover, RecInterpreter [59] examines the capacity of LLMs to decipher the representation space of sequential recommenders with sequence-recovery and -residual tasks.

However, the potential of LLMs cannot be utilized solely through simple ICL (e.g., [49]). While fine-tuning LLMs for recommendation shows promising results, it is constrained by the computational demands and availability of open-source LLMs. Instead, we introduce a new paradigm for ISR by automatically optimizing prompts, which efficiently guides LLMs to semantically comprehend the varying user intents within a session, thereby enhancing the accuracy and understandability of SR.

3 THE PROPOSED METHOD

This section introduces the PO4ISR paradigm, a simple yet powerful framework inspired by the work [38] in the area of NLP. It is specifically designed to efficiently guide LLMs in comprehending

varying user intents at a semantic level, with the goal of enhancing the accuracy and transparency of SR.

Framework Overview. Figure 2 illustrates the overall framework, mainly composed of three key components. Specifically, Prompt Initialization (*PromptInit*) is tasked with generating an initial prompt that directs LLMs in dynamically comprehending semantic user intents at the session level within a session. Subsequently, Prompt Optimization (*PromptOpt*) aims to evaluate, refine, augment, and optimize the initial prompt through self-reflection (i.e., inferring reasons from the collected error cases). Lastly, Prompt Selection (*PromptSel*) is designed to properly select optimized prompts by exploiting the robust generalizability of LLMs across diverse domains, thus maximizing the accuracy enhancements of SR.

3.1 Prompt Initialization (*PromptInit*)

Given a session, we first create an initial prompt for the task description. It seeks to guide LLMs in understanding the varying user intents at the semantic level, which thus empowers LLMs to make more accurate and comprehensible recommendations. The task description is demonstrated in Prompt 1 which divides the SR task into four subtasks by using the planning strategy [67]. Then, Prompt 1 is used to guide ChatGPT² to predict the next item based on the historical (training) user sessions fed by Prompt 2. For ease of understanding, we take one training session as an example, that is, the current session interactions: [1."Zenana Women's Cami Sets", 2."Monster Tattoos", 3."I Love You This Much Funny T-rex Adult T-shirt", 4."Breaking Bad Men's Logo T-Shirt", 5."Sofia the First Sofia's Transforming Dress", 6."Lewis N. Clark 2-Pack Neon Leather Luggage Tag", 7."Russell Athletic Women's Stretch Capri", 8."US Traveler New Yorker 4 Piece Luggage Set Expandable", 9."Soffe Juniors Football Capri"]. The target (ground truth) item "It's You Babe Mini Cradle, Medium" is ranked at position 19 out of 20 items

²Without a further statement, it is based on GPT-3.5-turbo.

in the candidate set. After using the initial prompt, the target item holds the 16th position in the re-ranked candidate set.

3.2 Prompt Optimization (PromptOpt)

PromptOpt strives to evaluate, refine, augment, and optimize the initial task description prompt with an iterative self-reflection. The detailed process is elaborated in what follows.

Collecting Error Cases. We randomly sample a batch (N_t) of sessions from training sessions (N) and guide ChatGPT to predict the next items with the initial prompts. Afterwards, we evaluate the recommendation outcomes, considering sessions where the target item ranks in the bottom half of the re-ranked candidate set as error cases. Following this rule, the example session mentioned in Section 3.1 is an error case, since the ranking position of the target item by using the initial prompt is 16 out of 20 candidates. These error cases, indicating that the corresponding prompts do not effectively guide LLMs in performing the SR task, will be utilized to further refine the prompts. We use N_e ($0 \leq N_e \leq N_t$) to denote the total number of such error cases for each batch.

Inferring Reasons. Understanding the reasons behind these collected error cases would greatly aid in refining the prompts, consequently enhancing recommendation performance. Thus, we leverage the self-reflection ability of ChatGPT, that is, asking ChatGPT to reconsider and offer justifications rooted in the identified errors. Inspired by [38], we adopt Prompt 3 to generate N_r reasons for each of the N_e error cases, where the ‘*error_case*’ includes the user session and candidate item set. Below demonstrates the generated reasons for one error case.

- One reason why the prompt could have gotten these examples wrong is that it assumes that the user’s interactive intent can be accurately inferred based solely on the items within each combination. However, the prompt does not provide any information about the user’s preferences, tastes, or previous interactions. Without this context, it is difficult to accurately infer the user’s intent from the items alone.
- Another reason is that the prompt does not specify how the combinations of items within the session should be discovered. It assumes that the combinations are already known and provided as input. However, in real-world scenarios, discovering meaningful combinations of items from a user’s session interactions can be a complex task. The prompt does not provide any guidance on how to perform this discovery process, which can lead to incorrect results.

Refining Prompts. With the inferred N_r reasons for each error case, we now ask ChatGPT to refine the current prompt accordingly using Prompt 4. One example of the refined prompt is illustrated as Prompt 5. By comparing the initial task description (Prompt 1) and the refined Prompt 5, we can easily note that the initial prompt is improved by considering two aspects: (1) user preference and taste and (2) the definition of item combinations. These two aspects are exactly consistent with the inferred reasons by ChatGPT.

Augmenting Prompts. With the refined prompts, we further ask ChatGPT to augment prompts (Prompt 7 is an example of augmentation) with the same semantic meanings using Prompt 6. Accordingly, for the N_e error cases, we finally obtain $2N_e$ improved prompts through refinement and augmentation. These prompts

will be further utilized for an iterative optimization based on their recommendation performance, as introduced in what follows.

Prompt 1: Task Description

Based on the user’s current session interactions, you need to answer the following subtasks step by step:

- 1 *Discover combinations of items within the session, where the size of combinations can be one or more.*
- 2 *Based on the items within each combination, infer the user’s interactive intent for each combination.*
- 3 *Select the intent from the inferred ones that best represent the user’s current preferences.*
- 4 *Based on the selected intent, please rerank the items in the candidate set according to the possibility of potential user interactions and show me your ranking results with the item index.*

Note that the order of all items in the candidate set must be provided, and the items for ranking must be within the candidate set.

Prompt 2: Input Data

Current session interactions: {[idx:“item title”, …]}

Candidate item set: {[idx:“item title”, …]}

Prompt 3: Inferring Reasons for Errors

*I’m trying to write a zero-shot recommender prompt.
My current prompt is {prompt}.*

But this prompt gets the following example wrong: {error_case}, give {Nr} reasons why the prompt could have gotten this example wrong.

Wrap each reason with <START> and <END>.

Prompt 4: Refining Prompts with Reasons

*I’m trying to write a zero-shot recommender prompt.
My current prompt is {prompt}.*

But this prompt gets the following example wrong: {error_case}.

Based on the example the problem with this prompt is that {reasons}.

Based on the above information, please write one improved prompt. The prompt is wrapped with <START> and <END>. The new prompt is:

Evaluating Prompts. From the pool of $2N_e$ prompts, identifying the most efficient ones with the best recommendation accuracy is crucial. One greedy way is to evaluate their performance with all historical user sessions. However, this may be quite computationally expensive. To improve the efficiency, we employ the upper confidence bound (UCB) Bandits [38] to efficiently estimate the performance. In particular, it iteratively samples one prompt based on its estimated performance and then evaluates the prompt on

a random batch of training sessions (N_t), and finally updates its performance based on the observed performance. The process is depicted by Algorithm 1, where the reward is calculated by the NDCG measuring the ranking position of the target item; and γ is the exploration parameter. With UCB Bandits, we can quickly obtain the estimated performance of the $2N_e$ prompts.

Prompt 5: The Refined Prompt

Given the user's current session interactions, you need to answer the following subtasks step by step:

- 1 Identify any patterns or relationships between the items within the session.
- 2 Based on the identified patterns, infer the user's interactive intent within each combination of items.
- 3 Consider the user's preferences, tastes, or previous interactions to select the intent that best represents their current preferences.
- 4 Rerank the items in the candidate set according to the likelihood of potential user interactions. Provide the ranking results with the item index.

Ensure that the order of all items in the candidate set is given, and the items for ranking are within the candidate set.

Prompt 6: Augmenting Prompts

Generate a variation of the following prompt while keeping the semantic meaning.

Input: {refined_prompt}.

Output:

Prompt 7: The Augmented Prompt

Please follow these steps to answer the subtasks based on the user's current session interactions:

- 1 Analyze the session items to find any patterns or relationships.
- 2 Use the identified patterns to determine the user's interactive intent for each combination of items.
- 3 Take into account the user's preferences, tastes, or previous interactions to choose the intent that best represents their current preferences.
- 4 Rank the items in the candidate set according to the likelihood of potential user interactions. Provide the ranking results along with the item index.

Make sure to include all items in the candidate set and only rank items within the candidate set.

Iterative Optimization. Based on UCB Bandits, we then iteratively optimize the prompts [38]. According to the estimated performance R , we select the Top- N_o prompts and carry these promising prompts forward to the subsequent iteration. Specifically, we first use the Top- N_o prompts to replace the prompts in the previous iteration, and then conduct the series of tasks, i.e., collecting error cases, inferring reasons, refining, augmenting, and evaluating prompts,

Algorithm 1: UCB-BANDITS

```

Input: prompt set  $\mathcal{P}$ , training session set  $\mathcal{S}$ , sampled session size  $N_t$ , maximum epoch  $E_1$ , reward function  $f(\cdot)$ ;
Output: the estimated performance  $R[\mathcal{P}]$ ;
// Initialization
1 for each  $p_i \in \mathcal{P}$  do
2    $R[p_i] = 0$ ; // initial estimated performance of  $p_i$ 
3    $S[p_i] = 0$ ; // initial frequency of  $p_i$  being evaluated
4 for  $e_1 = 1; e_1 \leq E_1; e_1 + +$  do
5   Sample  $p_i \leftarrow \arg \max_p \left( R[p] + \gamma \sqrt{\frac{\log(e_1)}{S[p]}} \right)$ ;
6   Randomly sample  $\mathcal{S}_t \subset \mathcal{S}$  where  $|\mathcal{S}_t| = N_t$ ;
7    $r[p_i] = 0$ ; // the initial accumulated reward
8   for each  $s \in \mathcal{S}_t$  do
9      $r[p_i] \leftarrow r[p_i] + f(p_i, s)$ ; // evaluate  $p_i$  with  $f(\cdot)$ 
// Update evaluation frequency and performance
10   $S[p_i] \leftarrow S[p_i] + N_t$ ;
11   $R[p_i] \leftarrow R[p_i] + \frac{r[p_i]}{N_t}$ ;
12 return  $R[\mathcal{P}]$ ;

```

as described in Algorithm 2. This iterative loop fosters incremental enhancements and exploration among various promising prompt candidates. Prompt 8 is an example of an optimized prompt with the best-estimated performance in the final iteration. After applying it on the example session mentioned in Section 3.1, the target item has been re-ranked to the 11th position, advancing eight positions from its previous placement in the candidate set.

Prompt 8: The Optimized Prompt

Please follow these steps to answer the given subtasks:

- 1 Analyze the combinations of items in the user's session, considering any patterns or criteria.
- 2 Deduce the user's interactive intent within each combination, taking into account their previous interactions and preferences.
- 3 Determine the most representative intent from the inferred ones that aligns with the user's current preferences.
- 4 Reorder the items in the candidate set based on the selected intent, considering potential user interactions.

Please provide the ranking results with item index.

Remember to provide the order of all items in the candidate set and ensure that the items for ranking are within the candidate set. Take into consideration the relevance of the items in the current session interactions to the candidate set, and incorporate the user's preferences and history into the recommendations.

3.3 Prompt Selection (PromptSel)

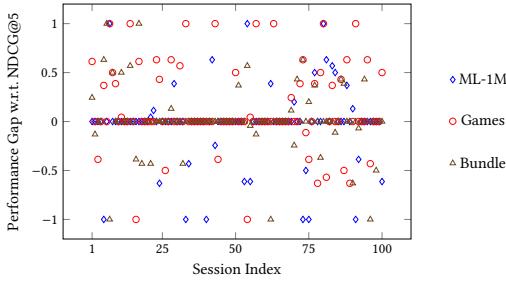
At the end of the iterative optimization, we have the Top- N_o prompts. Accordingly, one straightforward way is to choose the Top-1 prompt as the final selection due to its superior overall performance. However, we notice that although a majority of sessions achieve their peak accuracy with the Top-1 prompt, a subset of them displays the best performance with other Top prompts. This can be verified by Figure 3, which shows the performance gap between the Top-1 and Top-2 prompts on the validation sessions across three real-world

Algorithm 2: ITERATIVE-OPTIMIZATION

```

Input:  $\mathcal{S}, N_t, N_o, E_1, E_2, f(\cdot)$ ;
Output: the optimized prompt set  $\mathcal{P}_o$ ;
1  $\mathcal{E} \leftarrow \emptyset, \mathcal{P}_o \leftarrow p_{\text{init}}$ ;
2 for  $e_2 = 1; e_2 \leq E_2; e_2 + +$  do
3    $\tilde{\mathcal{P}} \leftarrow \emptyset$ ;
4   for each  $p_i \in \mathcal{P}_o$  do
5     Randomly sample  $\mathcal{S}_t \subset \mathcal{S}$  where  $|\mathcal{S}_t| = N_t$ ;
6     for each  $s \in \mathcal{S}_t$  do
7       if  $s$  is an error case then
8          $\mathcal{E} \leftarrow \mathcal{E}.append(s)$ ; // collect error cases
9     for each  $s \in \mathcal{E}, |\mathcal{E}| = N_e$  do
10      Generate  $N_r$  reasons; // infer reasons
11       $p_{ir} \leftarrow p_i \& N_r$  reasons; // refine prompt
12       $p_{ia} \leftarrow p_{ir}$ ; // augment prompt
13       $\tilde{\mathcal{P}}.append(p_{ir}, p_{ia})$ ;
14    $R[\tilde{\mathcal{P}}] \leftarrow \text{UCB-BANDITS}(\tilde{\mathcal{P}}, \mathcal{S}, N_t, E_1, f(\cdot))$ ; // evaluate prompts
15    $\mathcal{P}_o \leftarrow \text{Top-}N_o$  of  $\tilde{\mathcal{P}}$  based on  $R[\tilde{\mathcal{P}}]$ ; // update prompts
16 return  $\mathcal{P}_o$ ;

```

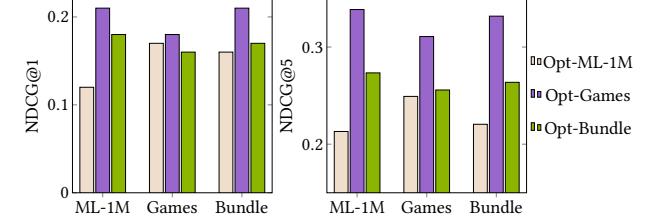
**Figure 3: Performance of Top-1&-2 prompts (same domain).**

datasets in the domains of movie, games, and e-commerce (with bundled products)³. From the results, it's evident that there are data points positioned where the gap is less than zero, signifying that the Top-2 prompt surpasses the Top-1 prompt in some cases. Thus, one can come up with potential solutions: (1) ensemble all top prompts to get the compensation results; and (2) train a classifier to select the best top prompt for each session. Nevertheless, we empirically discovered that (1) fails to yield promising results because the enhancements in the subset do not offset the declines in the majority; while (2) heavily relies on the accuracy of the classifier, thus bringing in extra uncertainty to the final performance.

Fortunately, the robust generalizability of LLMs inspires us to explore the cross-domain performance of these optimized prompts. Specifically, Figure 4 depicts the performance of the Top-1 prompts from the three domains across the three datasets, for instance, ‘Opt-Games’ (in purple) refers to the Top-1 prompt in the games domain. From the figure, we can easily notice that Opt-Games consistently performs the best not only in its games domain but also in the other two domains. One possible explanation could be attributed to the Games dataset having the shortest average session length (refer to the ‘Avg. Session Length’ in Table 1) alongside a moderate level of sparsity (see ‘Density Indicator’ in Table 1). These factors collectively alleviate the challenge of identifying the optimal prompt to capture crucial and unique information, thereby

³The details of the datasets utilized in our study are introduced in Section 4.1.1.**Table 1: Statistics of datasets. ‘Density Indicator’ refers to the average frequency of each item appearing in the dataset, calculated as (#Sessions × Avg. Session Length)/#Items.**

	#Items	#Sessions	Avg. Session Length	Density Indicator
ML-1M	3,416	784,860	6.85	1573.86
Games	17,389	100,018	4.18	24.04
Bundle	14,240	2,376	6.73	1.12

**Figure 4: Performance of Top-1 prompt (cross-domain).**

enhancing the overall performance of SR. Accordingly, we select the Opt-Games as the final prompt for all domains, the efficacy of which is verified in Section 4.2.

4 EXPERIMENTS AND RESULTS

We conduct extensive experiments to answer five research questions⁴. **(RQ1)** Does PO4ISR outperform baselines? **(RQ2)** How do different components affect PO4ISR? **(RQ3)** How do essential parameters affect PO4ISR? **(RQ4)** How does PO4ISR provide comprehensible SR? **(RQ5)** Is there any limitation of PO4ISR?

4.1 Experimental Setup

4.1.1 Datasets. We use three real-world datasets from various domains. In particular, MovieLen-1M (ML-1M)⁵ contains users' ratings of movies. Games is one subcategory from the Amazon dataset [34], containing users' ratings towards various video games. Bundle [47] contains session data for three subcategories (Electronic, Clothing, and Food) of Amazon, where the intents for each session are explicitly annotated via crowdsourcing workers. Table 1 shows the statistics of the datasets. For ML-1M and Games, we chronologically order the rated items into a sequence for each user and then divide it into sessions by day. For each dataset, we split the sessions into training, validation, and test sets with a ratio of 8:1:1, i.e., 80% of the initial sessions are treated as training sets; the subsequent 10% as the validation set; and the final 10% as the test set.

4.1.2 Baselines. We compare our proposed PO4ISR with ten baselines, which can be classified into three types. *The first type is the conventional methods.* **Mostpop** recommends the most popular items; **SKNN** [22] recommends session-level similar items; and **FPMC** [42] is the matrix factorization method with the first-order Markov chain. *The second type is the deep learning-based methods, which can be categorized into single-intent and multi-intent based ones.* **NARM** [26] is an RNN-based model with the attention mechanism to capture the main purpose from the hidden states; **STAMP** [32] learns the general intention of users by emphasizing the effect of the last item in the context; **GCN-GNN** [51] uses both

⁴Our code and data are available at <https://github.com/llm4sr/PO4ISR>⁵<https://grouplens.org/datasets/movielens/>

Table 2: Performance comparison on all datasets, where the best and runner-up results are highlighted in bold and marked by ‘’, ‘-’ means a very small value; and ‘Improve’ indicates the relative improvements comparing the best and runner-up results.**

Data	Metrics	Conventional Methods			Single-Intent Methods			Multi-Intent Methods			LLM Methods		Improve	<i>p</i> -value
		MostPop	SKNN	FPMC	NARM	STAMP	GCE-GNN	MCPRN	HIDE	Atten-Mixer	NIR	PO4ISR		
ML-1M	HR@1	0.0004	0.1270	0.1132	0.1692*	0.1584	0.1312	0.1434	0.1498	0.1490	0.0572	0.2000	18.20%	$4.3e^{-3}$
	HR@5	0.0070	0.3600	0.3748	0.5230*	0.5078	0.4748	0.4788	0.4998	0.4932	0.2326	0.5510	5.35%	$5.9e^{-2}$
	NDCG@1	0.0004	0.1270	0.1132	0.1692*	0.1584	0.1312	0.1434	0.1498	0.1490	0.0572	0.2000	18.20%	$4.3e^{-3}$
	NDCG@5	0.0053	0.2530	0.2464	0.3501*	0.3367	0.3044	0.3157	0.3256	0.3216	0.1436	0.3810	8.83%	$3.5e^{-3}$
Games	HR@1	-	0.0020	0.0498	0.0572	0.0556	0.0692	0.0522	0.0696	0.0530	0.1168*	0.2588	121.58%	$6.4e^{-5}$
	HR@5	-	0.0020	0.2564	0.2574	0.2586	0.2744	0.2416	0.2694	0.2472	0.3406*	0.5866	72.23%	$3.5e^{-5}$
	NDCG@1	-	0.0020	0.0498	0.0572	0.0556	0.0692	0.0522	0.0696	0.0530	0.1168*	0.2588	121.58%	$6.4e^{-5}$
	NDCG@5	-	0.0020	0.1508	0.1534	0.1555	0.1701	0.1432	0.1662	0.1475	0.2310*	0.4313	86.71%	$7.3e^{-6}$
Bundle	HR@1	-	-	0.0398	0.0322	0.0365	0.0360	0.0360	0.0458	0.0525	0.0975*	0.1697	74.05%	$2.0e^{-5}$
	HR@5	0.0042	-	0.2475	0.2322	0.2352	0.2237	0.2352	0.2585	0.2644	0.2832*	0.4328	52.82%	$2.6e^{-4}$
	NDCG@1	-	-	0.0398	0.0322	0.0365	0.0360	0.0360	0.0458	0.0525	0.0975*	0.1697	74.05%	$2.0e^{-5}$
	NDCG@5	0.0021	-	0.1395	0.1303	0.1339	0.1267	0.1490	0.1495	0.1549	0.1939*	0.3040	56.78%	$3.4e^{-5}$

local and global graphs to learn item representation thus obtaining the main intent of the session; **MCPRN** [50] models users’ multiple purposes to get the final session representation; **HIDE** [29] splits the item embedding into multiple chunks, with each chunk representing a specific intention to learn diverse intentions within context; and **Atten-Mixer** [65]: learns multi-granularity consecutive user intents to generate more accurate session representations. *The last type is the LLM-based method.* **NIR** [49] adopts zero-shot prompting for the next item recommendation.

4.1.3 Parameter Settings. We use Optuna (optuna.org) to automatically find out the optimal hyperparameters of all methods with 50 trials. In particular, the maximum training epoch is set as 100 with the early stop mechanism. The search space for batch size, item embedding size, and learning rate are {64, 128, 256}, {32, 64, 128} and { 10^{-4} , 10^{-3} , 10^{-2} }, respectively. For SKNN, K is searched from {50, 100, 150}. For NARM, the hidden size and layers are searched in [50, 200] stepped by 50 and in {1, 2, 3}, respectively. For GCE-GNN, the number of hops, and the dropout rate for global and local aggregators are respectively searched in {1, 2}, [0, 0.8] stepped by 0.2, and {0, 0.5}. For MCPRN, τ and the number of purpose channels are separately searched in {0.01, 0.1, 1, 10} and {1, 2, 3, 4}. For HIDE, the number of factors is searched in {1, 3, 5, 7, 9}; the regularization and balance weights are searched in { 10^{-5} , 10^{-4} , 10^{-3} , 10^{-2} }; the window size is searched in [1, 10] stepped by 1; and the sparsity coefficient is set as 0.4. For Atten-Mixer, the intent level L and the number of attention heads are respectively searched in [1, 10] stepped by 1 and in {1, 2, 4, 8}. For PO4ISR, $N = 50$, $N_t = 32$, $N_r = 2$, $N_o = 4$, $E_1 = 16$, $E_2 = 2$; and we randomly select 8 prompts as the input of the UCB Bandits in each iteration.

4.1.4 Evaluation Metrics. Following state-of-the-arts [49, 50, 61], HR@K and NDCG@K are adopted as the evaluation metrics. We set $K = 1/5$ since most users tend to prioritize the quality of items appearing at the top positions in real scenarios [31]. Generally, higher metric values indicate better ranking results. For a fair comparison, the candidate size of all methods is set as 20 following NIR [49]; all non-LLM baselines are trained with 150 randomly sampled sessions from the training set, while PO4ISR uses the subset of 50 sessions. This is because although PO4ISR uses 50 sessions to optimize prompts in each domain, it considers the performance

from (three) diverse domains for the final prompt selection. For non-LLM baselines, it is non-trivial to use the cross-domain performance to select the best models. This also helps showcase the superior advantages of LLMs on generalization. To manage the API call cost, for ML-1M and Games, we randomly sample 1000 sessions from the validation set and test set, respectively. For robust performance, we repeat the test procedure five times where each time we set different seed values (i.e., 0, 10, 42, 625, and 2023) to generate different candidate sets. Finally, we report the average results as presented in Table 2.

4.2 Results and Analysis

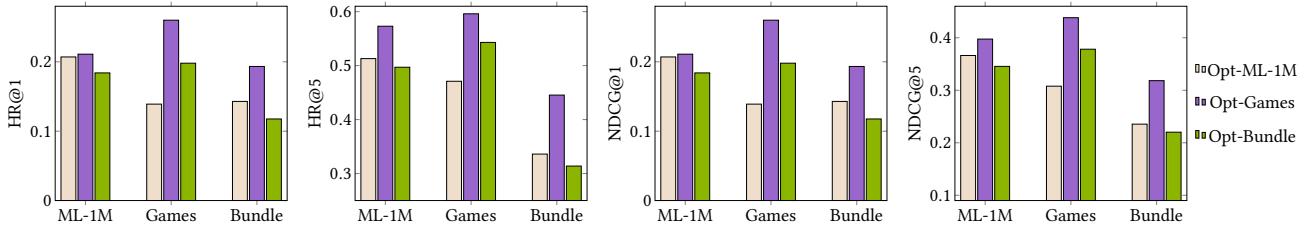
4.2.1 Overall Comparison (RQ1). Table 2 shows the performance of all methods on the three datasets. Several findings are noted.

(1) Regarding conventional methods (CMs), the model-based FPMC performs the best on sparser datasets Games and Bundles, while is slightly defeated by SKNN on the denser dataset ML-1M (see ‘Density Indicator’ in Table 1). This exhibits the strong capability of model-based methods in learning sequential patterns with sparse data. **(2)** For single-intent methods (SIMs), each gains its best performance on different datasets, e.g., NARM excels on ML-1M, whereas GCE-GNN wins on Games. Generally, GCE-GNN showcases advantages on shorter sessions with sparser datasets (e.g., Games). This is because such sessions do not contain sufficient local information, thus requiring global information (graph) to compensate. Conversely, NARM performs well on longer sessions with denser datasets (e.g., ML-1M), since such sessions possess substantial information (e.g., frequent patterns) making the fusion of global information potentially noisy. **(3)** Concerning multi-intent methods (MIMs), MCPRN displays the poorest performance across all datasets, as limited training data hinders the learning of proper parameters for multiple RNN channels. Besides, HIDE exceeds Atten-Mixer on relatively denser datasets (ML-1M and Games) but lags on the sparsest Bundle. **(4)** Among LLM-based methods (LLMMs), PO4ISR outperforms NIR, corroborating our prior assertion that basic in-context learning fails to fully exploit LLMs, thereby emphasizing the superiority of our prompt optimization paradigm.

Overall, CMs underperform SIMs or MIMs, underscoring the importance of capturing user intents for improved SR. The victory of MIMs over SIMs on Bundle validates the effectiveness of

Table 3: The results of ablation study across all datasets on the test set (seed 0).

	ML-1M					Games				Bundle				
	Initial	Top-1	EnSame	EnCross	PO4ISR	Initial	EnSame	EnCross	PO4ISR	Initial	Top-1	EnSame	EnCross	PO4ISR
HR@1	0.1430	0.2070	0.1120	0.1070	0.2110	0.0790	0.1540	0.1090	0.2600	0.0504	0.1176	0.0294	0.0840	0.1933
HR@5	0.4150	0.5130	0.4130	0.4250	0.5730	0.3510	0.5250	0.4410	0.5960	0.2437	0.3193	0.1891	0.2689	0.4454
NDCG@1	0.1430	0.2070	0.1120	0.1070	0.2110	0.0790	0.1540	0.1090	0.2600	0.0504	0.1176	0.0294	0.0840	0.1933
NDCG@5	0.2823	0.3662	0.2693	0.2640	0.3975	0.2110	0.3574	0.2779	0.4381	0.1396	0.2202	0.1119	0.1745	0.3183

**Figure 5: Performance of Top-1 prompt from each domain in the cross-domain scenario on the test set (seed 0).**

multi-intent learning. However, some of the SIMs (e.g., NARM) surpass MIMs (e.g., MCPRN) on ML-1M and Games, suggesting that fixed numbers of intents might constrain the capability of MIMs. LLMMs exhibit strength on sparser datasets compared to denser ones (e.g., NIR on Games vs. ML-1M), and shorter sessions compared to longer ones (e.g., NIR on Games vs. Bundles). Similar trends are also observed in PO4ISR, where the average improvement (12.65%) on ML-1M is smaller than that (100.55%) on Games. This demonstrates the advanced ability of LLMs to address the data sparsity issue. Lastly, our PO4ISR consistently achieves the best performance among all baselines, with an average improvement of 57.37% and 61.03% on HR and NDCG, respectively.

4.2.2 Ablation Study (RQ2). To verify the efficacy of different components, we compare PO4ISR with its four variants. In particular, ‘Initial’ means we only use the initial prompt without optimization for each domain; ‘Top-1’ means we merely adopt the Top-1 prompt within each domain; ‘EnSame’ means we ensemble the ranking results of both Top-1 and -2 prompts within each domain; ‘EnCross’ means we ensemble the ranking results of Top-1 prompts across all domains. The performance is shown in Table 3. Three observations are noted: (1) Initial performs worse than Top-1, which indicates the effectiveness of iterative prompt optimization; (2) both EnSame and EnCross underperform Top-1, implying that simply ensembling the top prompts either within the same domain or across different domains cannot efficiently improve the performance; and (3) PO4ISR with the Top-1 prompt in the games domain consistently achieves the best performance across all datasets, showcasing the efficacy of our cross-domain prompt selection strategy. This is further confirmed by Figure 5, presenting the performance of Top-1 prompt from each domain in the cross-domain scenario.

4.2.3 Parameter Analysis (RQ3). We further investigate the impact of important hyperparameters on PO4ISR. First, we examine the impact of different initial prompts. To this end, we substitute the initial task description (Prompt 1) with Prompt 9 incorporating two major changes: (1) we use ‘preferences’ to replace ‘intentions’, and the two terms have the same meanings in the context of ISR, and (2) we simplify the four subtasks into two subtasks. The results are

presented in Table 4 (rows 1-3 vs. rows 4-6), where we note that directly using Prompt 9 achieves better performance than using Prompt 1; meanwhile, the corresponding Top-1 prompt optimized based on Prompt 9 outperforms the Top-1 prompt optimized based on Prompt 1. This indicates that (1) simplified descriptions and subtasks division can improve the quality of the initial prompt, *vice versa*; (2) the quality of the initial prompts positively affects the final performance; (3) regardless of the quality of initial prompts, they can be largely enhanced with iterative optimization; (4) the lower-quality initial prompt yields larger overall improvements; and (5) the performance of PO4ISR presented in Table 2 is not the upper bound, and can be further improved with better initial prompts, showcasing its great potential. Besides, we study the impact of batch size N_t on the final performance by varying its values in {16, 32}, and the results are presented in Table 4 (row 2 vs. row 7). Accordingly, we find that $N_t = 32$ is the optimal setting.

Prompt 9: Task Description

Based on the user’s current session interactions, you need to answer the following tasks:

- 1 *Please infer the user’s preferences, considering that the user may have one or multiple preferences.*
- 2 *Based on inferred preferences, please rerank the items in the candidate set according to the possibility of potential user interactions and show me your ranking results with the item index.*

Note that the order of all items in the candidate set must be provided, and the items for ranking must be within the candidate set.

4.2.4 Visualization (RQ4). To illustrate the results generated by PO4ISR, we randomly sample one test session from Bundle as shown in Figure 6. Given the historical session, PO4ISR first detects four combinations of items, including $[i_1, i_2, i_8]$, $[i_3, i_4, i_6]$, $[i_5, i_7]$, $[i_9]$. The corresponding intents for each combination are (1) “this combination includes gummi bears, chamomile tea, and rosehip with hibiscus tea; the user might be interested in snacks and beverages

Table 4: Results of parameter analysis on Bundle (seed 0).

	HR@1	HR@5	NDCG@1	NDCG@5
Prompt 1 (Initial)	0.0504	0.2437	0.0504	0.1396
Top-1 ($N_t = 32$)	0.1176	0.3193	0.1176	0.2202
Improve	133.33%	31.02%	133.33%	57.74%
Prompt 9 (Initial)	0.1008	0.2479	0.1008	0.1706
Top-1 ($N_t = 32$)	0.1807	0.3739	0.1807	0.2744
Improve	79.27%	50.83%	79.27%	60.84%
Top-1 ($N_t = 16$)	0.1008	0.2857	0.1008	0.1860
Improve (vs. Prompt 1)	100%	17.23%	100%	33.24%

**Figure 6: The case study on Bundle.**

for relaxation or enjoyment"; (2) "this combination includes organic infant formula, organic baby food, and organic baby food pouches; the user's intent seems to be focused on organic and healthy options for infants"; (3) "this combination includes Sriracha hot sauce and green hot sauce; the user's intent appears to be related to spicy condiments or sauces" and (4) "this combination includes green superfood capsules; the user's intent might be to explore or maintain a healthy lifestyle". Then it identifies and outputs the most important intent: "based on the given combinations, the intent related to organic and healthy options for infants (Combination 2) seems to be the most specific and focused". Finally, it furnishes a re-ranked recommendation list, positioning the ground truth item "Earth's Best Organic Baby Food, Spaghetti With Cheese" (highlighted in a red-dot circle) from its initial 11th rank within the candidate set to the topmost position. In summary, the case study validates that PO4ISR can help provide more comprehensible recommendations.

4.2.5 Discussion on Hallucination (RQ5). Despite the success of PO4ISR on the SR task, it showcases limitations due to the inherent issue of LLMs. One issue is that PO4ISR may generate hallucination for some sessions (e.g., the response does not contain the ranking list or the ground truth item is not included in the ranking list), although we add hard constraints in the prompt such as "*the order of all items in the candidate set must be provided, and the items for ranking must be within the candidate set*". Table 5 illustrates the ratio

Table 5: Ratio of sessions with hallucination (seed 0).

	Opt-ML-1M	Opt-Games	Opt-Bundle	Average
ML-1M	0.30%	0.10%	0.30%	0.23%
Games	8.30%	6.80%	6.00%	7.03%
Bundle	7.56%	9.66%	10.92%	9.38%
Average	5.39%	5.52%	5.74%	5.55%

Table 6: Performance with and without JSON on Bundle.

	Opt-ML-1M		Opt-Games		Opt-Bundle	
	+JSON	-JSON	+JSON	-JSON	+JSON	-JSON
HR@1	0.0546	0.1429	0.0504	0.1933	0.0756	0.1176
HR@5	0.2227	0.3361	0.2185	0.4454	0.2395	0.3193
NDCG@1	0.0546	0.1429	0.0504	0.1933	0.0756	0.1176
NDCG@5	0.1336	0.2355	0.1289	0.3183	0.1531	0.2202
Ratio	7.14%	7.56%	8.82%	9.66%	9.66%	10.92%

of sessions having hallucinations on the test sets using the Top-1 prompt from each domain across the three datasets. Two major observations can be noted. (1) The ratio is at its lowest on ML-1M but peaks on Bundle (emphasized in blue). This discrepancy likely stems from ML-1M having the highest average repeat frequency of items across sessions, whereas Bundle exhibits the lowest trend (see Table 1). The frequent appearance of items in various sessions may simplify the pattern recognition process, thus reducing task complexity to some extent. Additionally, Bundle's diverse range of electronics, clothing, and food products elevates the complexity of the SR task compared to the more focused ML-1M dataset. (2) The optimal prompts from different domains show comparable performance as highlighted in pink. On average, there are around 5.55% sessions with hallucination, implying that further performance enhancements can be obtained by addressing this issue and exhibiting the latent potential of LLMs for SR (note that the HR and NDCG values are set as 0 for sessions with hallucination).

To alleviate the hallucination issue, we involve JSON mode⁶ in the response to better control the output by adding the constraint as: *Provide the ranking results for the candidate set using JSON format, following this format without deviation: ["Item ID": "correspond item index", "Item Title": "correspond Item Title"]*. Table 6 illustrates the performance contrast of PO4ISR on Bundle, with and without the JSON mode in the response, which indicates that employing JSON mode marginally reduces bad cases, yet significantly compromises recommendation accuracy. Contrarily, we find most such cases can be better eased by using GPT-4 with less compromised accuracy.

5 CONCLUSION

Inspired by the reasoning capability of LLMs, we introduce a new paradigm – PO4ISR – for intent-aware session recommendation. It aims to discover varying numbers of semantic intents hidden in different sessions for more accurate and comprehensible recommendations through iterative prompt optimization. Specifically, the Prompt Initialization module first creates the initial prompt to instruct LLMs to predict the next item by inferring varying intents reflected in a session. Then, the prompt optimization module is devised to optimize prompts with iterative self-reflection in an automatic manner. Finally, the prompt selection module seeks to

⁶<https://platform.openai.com/docs/guides/text-generation/json-mode>

appropriately select optimal prompts based on the robust generalizability of LLMs across diverse domains. Extensive experiments on real-world datasets show the superiority of PO4ISR against other counterparts. Furthermore, several insightful discoveries are made to guide subsequent studies in this area.

REFERENCES

- [1] Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. 2010. Best arm identification in multi-armed bandits.. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*. 41–53.
- [2] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. 2002. Sequential pattern mining using a bitmap representation. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 429–435.
- [3] Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Fuli Feng, Xiangnaan He, and Qi Tian. 2023. A bi-step grounding paradigm for large language models in recommendation systems. *arXiv preprint arXiv:2308.08434* (2023).
- [4] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: an effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*. 1007–1014.
- [5] Jingfan Chen, Guanghui Zhu, Haojun Hou, Chunfeng Yuan, and Yihua Huang. 2022. AutoGSR: Neural architecture search for graph-based session recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1694–1704.
- [6] Qian Chen, Zhiqiang Guo, Jianjun Li, and Guohui Li. 2023. Knowledge-enhanced Multi-View Graph Neural Networks for Session-based Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 352–361.
- [7] Qian Chen, Jianjun Li, Zhiqiang Guo, Guohui Li, and Zhiying Deng. 2023. Attribute-enhanced dual channel representation learning for session-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*. 3793–3797.
- [8] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1172–1180.
- [9] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten de Rijke. 2020. Improving end-to-end sequential recommendations with intent-aware diversification. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*. 175–184.
- [10] Minjin Choi, Jinhong Kim, Joonseok Lee, Hyunjung Shim, and Jongwuk Lee. 2022. S-Walk: accurate and scalable session-based recommendation with random walks. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM)*. 150–160.
- [11] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt’s capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*. 1126–1132.
- [12] Jiayian Guo, Yaming Yang, Xiangchen Song, Yuan Zhang, Yujing Wang, Jing Bai, and Yan Zhang. 2022. Learning multi-granularity consecutive user intent unit for session-based recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM)*. 343–352.
- [13] Jiayian Guo, Peiyian Zhang, Chaozhuo Li, Xing Xie, Yan Zhang, and Sunghun Kim. 2022. Evolutionary preference learning via graph nested gru o de for session-based recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*. 624–634.
- [14] Qilong Han, Chi Zhang, Rui Chen, Riwei Lai, Hongtao Song, and Li Li. 2022. Multi-faceted global item relation learning for session-based recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1705–1715.
- [15] 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)*. 1096–1102.
- [16] Zhanhui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Majumder, Nathan Kallus, and Julian McAuley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*. Just Accepted.
- [17] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*. 843–852.
- [18] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.
- [19] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*. 241–248.
- [20] Yupeng Hou, Binbin Hu, Zhiqiang Zhang, and Wayne Xin Zhao. 2022. Core: simple and effective session-based recommendation within consistent representation space. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1796–1801.
- [21] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845* (2023).
- [22] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys)*. 306–310.
- [23] Di Jin, Luzhi Wang, Yizhen Zheng, Guojie Song, Fei Jiang, Xiang Li, Wei Lin, and Shirui Pan. 2023. Dual intent enhanced graph neural network for session-based new item recommendation. In *Proceedings of the ACM Web Conference (TheWebConf)*. 684–693.
- [24] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. 197–206.
- [25] Siqi Lai, Erli Meng, Fan Zhang, Chenliang Li, Bin Wang, and Aixin Sun. 2022. An attribute-driven mirror graph network for session-based recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1674–1683.
- [26] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*. 1419–1428.
- [27] Jimming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023. GPT4Rec: A generative framework for personalized recommendation and user interest interpretation. *arXiv preprint arXiv:2304.03879* (2023).
- [28] Yinfeng Li, Chen Gao, Xiaoyi Du, Huazhou Wei, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Spatiotemporal-aware Session-based Recommendation with Graph Neural Networks. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*. 1209–1218.
- [29] Yinfeng Li, Chen Gao, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1997–2002.
- [30] Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2023. A Multi-facet Paradigm to Bridge Large Language Model and Recommendation. *arXiv preprint arXiv:2310.06491* (2023).
- [31] Hongyang Liu, Zhu Sun, Xinghua Qu, and Fuyong Yuan. 2021. Top-aware recommender distillation with deep reinforcement learning. *Information Sciences (INS)* 576 (2021). 642–657.
- [32] Qiao Liu, Yifu Wu, Refueo Mokhosi, and Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1831–1839.
- [33] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction (UMUAI)* 28 (2018), 331–390.
- [34] 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)*. 188–197.
- [35] Sejoon Oh, Ankur Bhardwaj, Jongseok Han, Sungchul Kim, Ryan A Rossi, and Sriyan Kumar. 2022. Implicit session contexts for next-item recommendations. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*. 4364–4368.
- [36] Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. 2022. Heterogeneous global graph neural networks for personalized session-based recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM)*. 775–783.
- [37] Andreas Peintner, Amir Reza Mohammadi, and Eva Zangerle. 2023. SPARE: Shortest path global item relations for efficient session-based recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*. 58–69.
- [38] Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with “gradient descent” and beam search. *arXiv preprint arXiv:2305.03495* (2023).
- [39] Shutong Qiao, Wei Zhou, Junhao Wen, Hongyu Zhang, and Min Gao. 2023. Bi-channel multiple sparse graph attention networks for session-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information*

- and Knowledge Management (CIKM)*. 2075–2084.
- [40] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. 579–588.
- [41] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys)*. 130–137.
- [42] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*. 811–820.
- [43] Scott Sammer, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. 2023. Large language models are competitive near cold-start recommenders for language-and-item-based preferences. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*. 890–896.
- [44] Wenzhuo Song, Shoujin Wang, Yan Wang, Kunpeng Liu, Xueyan Liu, and Ming-hao Yin. 2023. A counterfactual collaborative session-based recommender system. In *Proceedings of the ACM Web Conference (TheWebConf)*. 971–982.
- [45] Jiajie Su, Chaochao Chen, Weiming Liu, Fei Wu, Xiaolin Zheng, and Haoming Lyu. 2023. Enhancing hierarchy-aware graph networks with deep dual clustering for session-based recommendation. In *Proceedings of the ACM Web Conference (TheWebConf)*. 165–176.
- [46] 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 (CIKM)*. 1441–1450.
- [47] Zhu Sun, Jie Yang, Kaidong Feng, Hui Fang, Xinghua Qu, and Yew Soon Ong. 2022. Revisiting bundle recommendation: datasets, tasks, challenges and opportunities for intent-aware product bundling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2900–2911.
- [48] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 17–22.
- [49] Lei Wang and Ee-Peng Lim. 2023. Zero-Shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153* (2023).
- [50] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*.
- [51] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 169–178.
- [52] Zhidan Wang, Wenwen Ye, Xu Chen, Wenqiang Zhang, Zhenlei Wang, Lixin Zou, and Weidong Liu. 2022. Generative session-based recommendation. In *Proceedings of the ACM Web Conference (TheWebConf)*. 2227–2235.
- [53] Chunyu Wei, Bing Bai, Kun Bai, and Fei Wang. 2022. Gsl4rec: Session-based recommendations with collective graph structure learning and next interaction prediction. In *Proceedings of the ACM Web Conference (TheWebConf)*. 2120–2130.
- [54] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 33. 346–353.
- [55] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 35. 4503–4511.
- [56] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph contextualized self-attention network for session-based recommendation. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 19. 3940–3946.
- [57] Heeyoon Yang, YunSeok Choi, Gahyung Kim, and Jee-Hyong Lee. 2023. LOAM: Improving Long-tail Session-based Recommendation via Niche Wall Augmentation and Tail Session Mixup. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 527–536.
- [58] Liqi Yang, Linhan Luo, Lifeng Xin, Xiaofeng Zhang, and Xinni Zhang. 2022. DAGNN: Demand-aware graph neural networks for session-based recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.
- [59] Zhengyi Yang, Jiancan Wu, Yanchen Luo, Jizhi Zhang, Yancheng Yuan, An Zhang, Xiang Wang, and Xiangnan He. 2023. Large language model can interpret latent space of sequential recommenders. *arXiv preprint arXiv:2310.20487* (2023).
- [60] Ghim-Eng Yap, Xiao-Li Li, and Philip S Yu. 2012. Effective next-items recommendation via personalized sequential pattern mining. In *Proceedings of the 17th International Conference on Database Systems for Advanced Applications (DASFAA)*. 48–64.
- [61] Qing Yin, Hui Fang, Zhu Sun, and Yew-Soon Ong. 2023. Understanding diversity in session-based recommendation. *ACM Transactions on Information Systems (TOIS)* 42, 1 (2023), 1–34.
- [62] Dianer Yu, Qian Li, Hongzhi Yin, and Guandong Xu. 2023. Causality-guided graph learning for session-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*. 3083–3093.
- [63] Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2020. TAGNN: Target attentive graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1921–1924.
- [64] 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. In *The 1st Workshop on Personalized Generative AI @CIKM (PGAI)*.
- [65] Peiyan Zhang, Jiayan Guo, Chaozhuo Li, Yuezqi Xie, Jae Boum Kim, Yan Zhang, Xing Xie, Haohan Wang, and Sunghun Kim. 2023. Efficiently leveraging multi-level user intent for session-based recommendation via atten-mixer network. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM)*. 168–176.
- [66] Xiaokun Zhang, Bo Xu, Liang Yang, Chenliang Li, Fenglong Ma, Haifeng Liu, and Hongfei Lin. 2022. Price does matter! modeling price and interest preferences in session-based recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1684–1693.
- [67] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2023. Least-to-most prompting enables complex reasoning in large language models. In *Proceedings of the 17th International Conference on Learning Representations (ICLR)*.