Think Before Recommend: Unleashing the Latent Reasoning Power for Sequential Recommendation

Jiakai Tang[1*§], Sunhao Dai[1*], Teng Shi[1], Jun Xu[1], Xu Chen[1†], Wen Chen[2†], Jian Wu[2], Yuning Jiang[2]
[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[2]Alibaba Group, Beijing, China
{tangjiakai5704,sunhaodai,shiteng,junxu,xu.chen}@ruc.edu.cn
{chenyu.cw,joshuawu.wujian,mengzhu.jyn}@alibaba-inc.com

Abstract

Sequential Recommendation (SeqRec) aims to predict the next item by capturing sequential patterns from users' historical interactions, playing a crucial role in many real-world recommender systems. However, existing approaches predominantly adopt a direct forward computation paradigm, where the final hidden state of the sequence encoder serves as the user representation. We argue that this inference paradigm, due to its limited computational depth, struggles to model the complex evolving nature of user preferences and lacks a nuanced understanding of long-tail items, leading to suboptimal performance. To address this issue, we propose **ReaRec**, the first inference-time computing framework for recommender systems, which enhances user representations through implicit multi-step reasoning. Specifically, ReaRec autoregressively feeds the sequence's last hidden state into the sequential recommender while incorporating special reasoning position embeddings to decouple the original item encoding space from the multi-step reasoning space. Moreover, we introduce two lightweight reasoning-based learning methods, Ensemble Reasoning Learning (ERL) and Progressive Reasoning Learning (PRL), to further effectively exploit ReaRec's reasoning potential. Extensive experiments on five public real-world datasets and different SeqRec architectures demonstrate the generality and effectiveness of our proposed ReaRec. Remarkably, post-hoc analyses reveal that ReaRec significantly elevates the performance ceiling of multiple sequential recommendation backbones by approximately 30%-50%. Thus, we believe this work can open a new and promising avenue for future research in inference-time computing for sequential recommendation.

Keywords

Sequential Recommendation, Inference-time Reasoning

∗ Equal Contribution.
§ Work done during internship at Alibaba Group.
† Corresponding authors.
Our code will be available at https://github.com/TangJiakai/ReaRec.

SIGIR '25, Padua, Italy

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Jiakai Tang[1*§], Sunhao Dai[1*], Teng Shi[1], Jun Xu[1], Xu Chen[1†], Wen Chen[2†], Jian Wu[2], Yuning Jiang[2]. 2025. Think Before Recommend: Unleashing the Latent Reasoning Power for Sequential Recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Recommender systems (RS) have become ubiquitous in modern daily life, powering personalized services across domains such as e-commerce platforms [23, 42], music recommendation services [3, 41], and video streaming applications [14, 40]. To accurately capture a user's next interaction intent, sequential recommendation algorithms are designed to analyze historical interactions to mine underlying sequential patterns and model latent user preferences [1, 5, 32]. Current mainstream sequential recommendation models, such as SASRec [13] and UniSRec [10], adopt a Transformer-based architecture, leveraging their power attention mechanisms to adaptively weight past interacted items and use the final position's encoded output as the user representation, as illustrated in Fig. 1(a). However, we argue this prevailing direct forward inference paradigm may lack nuanced comprehension of dynamic user preferences and evolving interest patterns, leading to suboptimal modeling for long-tail user interest and unpopular items. Despite their efficiency, we argue that these direct inference paradigms often fall short in modeling long-tail users with fewer interactions and less popular items—scenarios that inherently demand more nuanced reasoning and deeper representation learning.

Recently, many studies from the natural language processing (NLP) community have demonstrated that *Chain-of-Thought (CoT)* during inference can significantly improve the performance of *Large Language Models (LLMs)* on complex tasks like mathematics and coding [9, 22, 27, 33]. By allowing the model to perform multi-step deliberation before generating a final output, CoT-based reasoning enhances the model's capacity to handle complex problems beyond what direct inference allows. Furthermore, Feng et al. [6] theoretically uncover that the emergent thinking capabilities are attributed to the increased computational depth introduced by CoT-based reasoning, which allows models to overcome the expressivity limitations of direct answer even with constrained parameter sizes.

Motivated by these insights, we explore whether a similar *think-before-action* paradigm can benefit sequential recommendation, especially for challenging cases such as long-tail users and items. We propose ***ReaRec***, a novel reasoning-enhanced framework that enables SeqRec models to engage in implicit multi-step reasoning

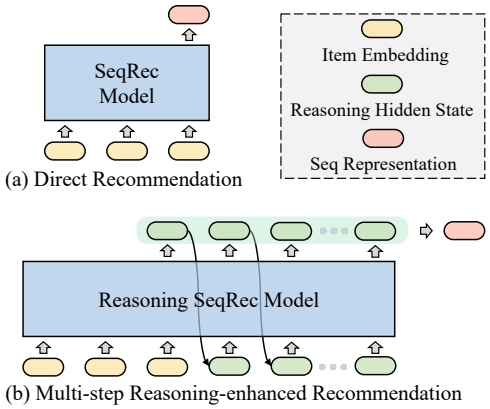

Figure 1: Illustration of traditional direct inference (*i.e.*, reasoning-free) and our proposed multi-step reasoning-enhanced sequential recommendation framework.

during inference. As shown in Fig. 1(b), ReaRec performs autoregressive reasoning over latent representations before producing the final user embedding, thereby deepening feature crossing and improving representational richness. To prevent the recommender from confusing the sequence encoding stage and reasoning stage, we design a specialized positional encoding scheme to explicitly distinguish item representations from reasoning inputs. However, unlike NLP tasks, where explicit reasoning chains naturally provide process supervision to guide model optimization [16, 18, 21], implicit reasoning in sequential recommendation lacks effective intermediate signals. This absence of stepwise guidance could lead to unpredicted *reasoning degradation* issues, causing the recommender to either replicate prior reasoning patterns or progressively drift away from accurately modeling the user's true interest distribution. Consequently, this may significantly impair the robustness and generalization capability of the recommendation model.

To address the aforementioned challenges, we propose two simple yet effective reasoning learning strategies, ***Ensemble Reasoning Learning (ERL)*** and ***Progressive Reasoning Learning (PRL)***, to fully exploit the reasoning power of our ReaRec framework. For the ERL method, it leverages the idea of *ensemble learning* to construct multi-order user representations to comprehensively capture latent interest distributions from diverse perspectives. Specifically, we introduce multi-step supervised optimization to alleviate the optimization difficulty in deep reasoning processes. Furthermore, to prevent reasoning-pattern degradation, we incorporate a representation diversity regularizer to mitigate output homogeneity in multi-step reasoning. For the PRL method, inspired by *curriculum learning*, we design a progressive temperature annealing mechanism to guide the model from initial exploitation to the gradual refinement of modeled sequential patterns. This approach enables the model to progressively learn the user's true interest distributions. Moreover, we also propose a reasoning-aware contrastive learning objective to enhance the reasoning robustness ability by simulating the error self-correction process, thus achieving better generalization performance.

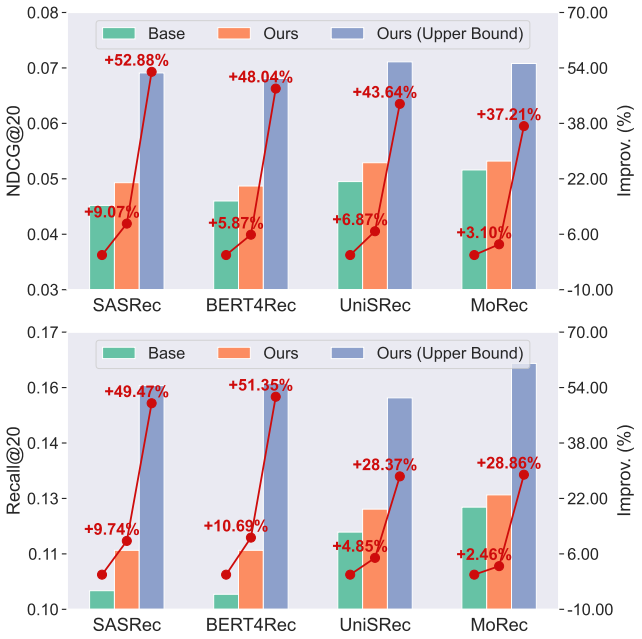

Figure 2: Empirical performance gains and potential upper bound analysis of optimal reasoning steps (K = 2) on Yelp dataset across different SeqRec models.

Our extensive experiments on five benchmark datasets demonstrate the effectiveness of the proposed ReaRec framework. In particular, the ReaRec achieves an average performance gain of 7.49% across all metrics while incurring only 3.51% additional inference latency (*cf.* Sec. 4.2 and Sec. 4.3.3). Moreover, further analysis reveals several interesting empirical findings: **(1) Enhancing modeling capability for underrepresented groups.** The multi-step reasoning process steadily enhances the recommendation quality of users with sparse interactions and long-tail items. **(2) Remarkable performance ceiling breakthrough.** Post-hoc optimal reasoning step analysis shows that our method elevates the performance ceilings for different backbone models by approximately 30%-50% (as shown in Fig. 2), highlighting its promising capability. We are optimistic that our proposed RecRec will open new avenues for exploring inference-time scaling for recommender systems.

Our main contributions are summarized as follows:

- We propose **ReaRec**, a novel reasoning-enhanced sequential recommendation framework that empowers SeqRec models to perform implicit multi-step reasoning during inference. To our knowledge, this is the first work to systematically explore inference-time computational power within recommender systems.
- We introduce two reasoning learning strategies, ERL and PRL, which leverage the ideas of ensemble learning and curriculum learning to efficiently optimize the implicit reasoning process and alleviate reasoning degradation issues.
- Extensive experiments on five real-world datasets and various representative SeqRec models validate the generality and effectiveness of ReaRec. Notably, our detailed post-hoc analysis reveals that ReaRec can significantly raise the performance ceiling, achieving significant improvements by up to 50%.

- We identify some challenges faced by current reasoning-enhanced recommendation methods and the future opportunities, stimulating a new research direction at the intersection of inference-time computing and sequential recommendation.

2 Preliminary

In this section, we formally define the sequential recommendation task and introduce the typical sequential recommendation pipeline.

2.1 Problem Definition

Formally, let $\mathcal{U}$ and $\mathcal{V}$ denote the sets of users and items, respectively, with $M = |\mathcal{U}|$ and $N = |\mathcal{V}|$ representing the number of users and items. For each user $u \in \mathcal{U}$, we define their chronological interaction sequence as $\mathcal{S}^u = [v_1^u, v_2^u, \ldots, v_{n_u}^u]$, where n_u represents the length of the interaction sequence $\mathcal{S}_u$. Each item $v \in \mathcal{V}$ has a unique ID and a set of textual attributes (such as title, product feature, and other side information). These attributes are stored in a dictionary $\mathcal{D}_v = \{k_1 : a_1, k_2 : a_2, \ldots, k_m : a_m\}$, where k_i and a_i represent the key and value of the i-th attribute, respectively. Here, m refers to the total number of attributes associated with item v. The overall text description for item v is constructed by concatenating its attributes in the format of an unordered list: "The item information is as follows: \n- k_1:a_1 \n- k_2:a_2 \n ... \n- k_m:a_m".

The goal of sequential recommendation is to predict the next item a user will interact with, based on historical interaction data. Given the interaction sequences for all users $\mathcal{S} = \{\mathcal{S}^{u_1}, \mathcal{S}^{u_2}, \ldots, \mathcal{S}^{u_M}\}$, where $\mathcal{S}^{u_i}$ represents the interaction sequence of user u_i, and $\mathcal{S}_{1:t}^{u_i} = [v_1^u, v_2^u, \ldots, v_t^u]$ denotes the first t interaction records of user u_i. Given the item embedding matrix $\mathbf{E} \in \mathbb{R}^{N \times d}$, where d is the dimension of the item embedding, the sub-sequence $\mathcal{S}_{1:t}^{u_i}$ is encoded to obtain the corresponding item embeddings $\mathbf{E}_{1:t}^{u_i} = [\mathbf{e}_{v_1^u}, \mathbf{e}_{v_2^u}, \ldots, \mathbf{e}_{v_t^u}]$. The recommender's learning objective is to maximize the prediction probability of the next item $v_{t+1}^{u_i}$ based on the historical interaction data, which is formally defined as

$$\max_{\Theta} \sum_{u \in \mathcal{U}} \sum_{t=1}^{n_u - 1} P(v_{t+1}^u | \mathcal{S}_{1:t}^u; \Theta), \tag{1}$$

where Θ denotes the parameters of the recommendation model.

2.2 Sequential Recommendation Pipeline

In a typical sequential recommendation pipeline, users' historical interactions are first encoded into item embeddings. These item embeddings are then fed into a sequential model (*e.g.*, transformer-based models) to produce a sequence representation, typically using the output from the final position (as illustrated in Fig. 1(a)). Finally, this sequence representation is used to calculate similarity scores with candidate item embeddings (such as dot product [39, 40] or cosine similarity[15, 35]) to predict the probability of the user's interaction with the next item.

In general, mainstream sequential recommendation methods can be broadly categorized into two main types, distinguished primarily by their approaches to encoding item representations:

(1) **ID-based Encoding**: The ID-based approach uses one-hot encoding for the item's discrete representation and retrieves the item's embedding from the embedding matrix. Representative sequential recommendation methods employing this encoding approach include SASRec [13], BERT4Rec[24], etc.

(2) **Text-based Encoding**: The text-based item representation usually involves feeding the item's string-formatted description into a pre-trained language model (such as BERT [19], LLaMA [8], etc.), and then utilizing average pooling or extracting hidden state from special positions (e.g., [CLS] or the last position) as the item's encoding [7, 15, 17]. Popular recommendation models utilizing text-based encoding include UniSRec [10], MoRec [38], etc.

In this paper, since the proposed reasoning framework is model-agnostic, we omit the details of how item representations are obtained and consistently use $\mathbf{e}_v \in \mathbf{E}$ to denote item v' representations.

3 Methodology

In this section, we introduce ReaRec, a novel, simple, and highly scalable recommendation framework designed to unleash a model's latent sequential reasoning capability. Instead of the traditional direct recommendation without reasoning, our approach leverages multi-step implicit reasoning to refine user representations, fully exploiting the computational potential of sequential models to approximate the true distribution of user interests.

In what follows, we first introduce ReaRec, our foundational framework for inference-time computation extension (Sec. 3.1). We then propose two lightweight methods—Ensemble Reasoning Learning (Sec. 3.2) and Progressive Reasoning Learning (Sec. 3.3)—to address the aforementioned challenges. The overall framework of ReaRec is illustrated in Fig. 3.

3.1 ReaRec Backbone

Our proposed ReaRec is model-agnostic and can be easily integrated into a variety of sequential recommenders. To better explain our work, we illustrate our framework using the widely adopted transformer [30] architecture in sequential recommendation tasks as an example, demonstrating how we extend computational capacity during inference with our backbone.

3.1.1 ***Self-attention Sequence Encoding.*** Given a user's historical sequence $\mathcal{S}_u = [v_1^u, v_2^u, \ldots, v_n^u]$, we can obtain the item embeddings of these n items by looking up the embedding matrix $\mathbf{E}$. To fully leverage sequential information, we inject *Absolute Position Embeddings* into the item embeddings at the input layer. Specifically, for a given item v at position i, the input representation is constructed by summing its item embedding $\mathbf{e}_v$ and the corresponding positional embedding $\mathbf{p}_i^I$:

$$\mathbf{h}_i^0 = \mathbf{e}_v + \mathbf{p}_i^I, \tag{2}$$

where $\mathbf{p}_i^I$ is obtained by looking up the learnable positional embedding matrix $\mathbf{P}^I \in \mathbb{R}^{n \times d}$. Next, we develop the item sequence encoder $f(\cdot)$ by stacking multiple multi-head self-attention layers (denoted as $MHSA(\cdot)$) and point-wise feed-forward networks (denoted as $FFN(\cdot)$) to capture the complicated sequence features:

$$\mathbf{H}^l = f(\mathbf{H}^{l-1}) = FFN(MHSA(\mathbf{H}^{l-1})), \tag{3}$$

where $\mathbf{H}^l = [\mathbf{h}_1^l, \mathbf{h}_2^l, \ldots, \mathbf{h}_n^l]$ denotes the concatenated hidden states at the l-th layer. In the conventional paradigm, the output at the last position of the final layer is directly used as the final user representation, *i.e.*, $\mathbf{h}_u = \mathbf{H}^L[-1]$, where L is the number of layers.

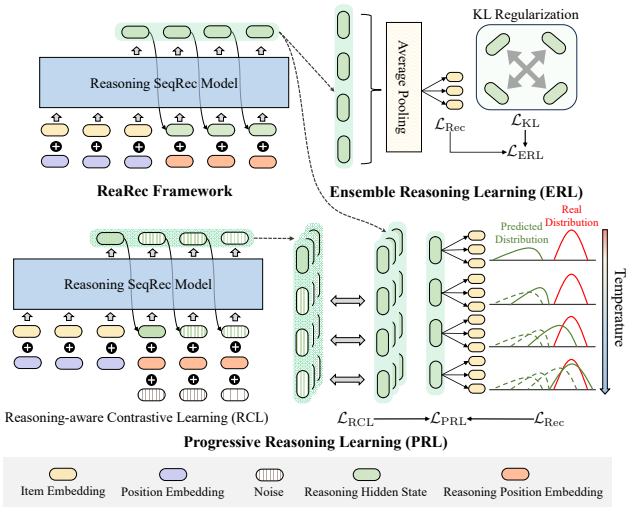

Figure 3: Overview of the proposed ReaRec framework and two reasoning-enhanced learning strategies: Ensemble Reasoning Learning and Progressive Reasoning Learning.

3.1.2 ***Extended Inference-Time Reasoning.*** Existing sequential recommenders that rely only on non-reasoning forward inference struggle to directly model item sequence patterns, fundamentally constrained by their limited computation power to capture nuanced user interest. To address this problem, we propose **implicit reasoning mechanism** to augment the computational capacity, enabling the enhanced refinement of user interest modeling to more precisely approximate real preference distributions.

Specifically, rather than directly using $\mathbf{H}^L[-1]$ as the user representation, we autoregressively feed the hidden state of the last position back into the encoder for ***K*-pass forward computations**. By effectively increasing inference-time computation, this approach further unleashes the model's potential to capture intricate sequential dependencies. However, this inference strategy deviates from the original objective of sequential recommendation models, namely next-item prediction. To bridge this task gap, we introduce the ***Reasoning Position Embedding (RPE)***, denoted as $\mathbf{P}^R \in \mathbb{R}^{K\times d}$, to distinguish between the sequence encoding phase and the reasoning phase. At the k-th reasoning step, the model's input embedding is defined as $\mathbf{H}^0 \in \mathbb{R}^{(n+k-1)\times d}$. The first n positions remain unchanged from the original input (*i.e.*, Eq. (2)), while the latent representation $\mathbf{h}^0_{n+i}$ at position $n+i$ is calculated as the summation of the last output $\mathbf{h}^L_{n+i-1}$ from the previous step and the i-th reasoning position embedding $\mathbf{p}^R_i$:

$$\mathbf{h}^0_{n+i} = \mathbf{h}^L_{n+i-1} + \mathbf{p}^R_i. \tag{4}$$

To differentiate between item encoding outputs and reasoning outputs, we denote the hidden states of the model's final layer from position n to $n+k$ as $\mathbf{R} = [\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_k]$, where $\mathbf{r}_i \in \mathbb{R}^d$ represents the reasoning hidden state at the i-th step. To obtain the user representation, a straightforward approach is to follow the traditional paradigm *i.e.*, use the last reasoning output $\mathbf{r}_K$ as $\mathbf{h}_u$. Then, we calculate the predicted probability for the user u as $\hat{y} = \text{softmax}(\mathbf{h}_u \cdot \mathbf{E}^\top)$ and use cross-entropy loss as the recommendation objective function:

$$\mathcal{L}_{\text{Rec}} = -\log \hat{y}_{v^+}, \tag{5}$$

where $\hat{y}_{v^+}$ denotes the prediction probability of the ground-truth item v^+ for user u's next interaction.

However, this naive optimization objective still faces a critical issue: the lack of supervision signals for intermediate reasoning states makes the model vulnerable to the risk of reasoning pattern degradation. Next, we introduce two simple yet effective reasoning learning strategies to address these challenges.

3.2 Ensemble Reasoning Learning (ERL)

To provide effective supervised signals for the implicit reasoning process, we propose an *Ensemble Reasoning Learning (ERL)* method. This approach uses the hidden states of different reasoning steps as multi-view representations of the user's evolving interests. In other words, we apply the idea of *ensemble learning* [4, 20] to aggregate diverse reasoning results from different reasoning steps, thereby avoiding suboptimal performance caused by the final output alone.

3.2.1 ***Multi-Step Reasoning Supervision.*** Specifically, we treat the reasoning hidden states from multiple steps as multi-vector user representations and apply cross-entropy loss (*cf.* Eq. (5)) to the ensembled sequence representation to enhance process guidance. Therefore, instead of using only the reasoning state at the last step, ERL utilizes an average pooling layer to aggregate the reasoning hidden states from all steps to obtain the final user representation, i.e., $\mathbf{h}_u = \frac{1}{K}\sum_{i=0}^{K}\mathbf{r}_i$.

3.2.2 ***KL Divergence Regularization.*** However, simply using the above recommendation objective for model training is obviously inefficient. The recommender may take shortcuts by directly copying the previous reasoning output to optimize the parameters, which can lead to a *pattern collpase effect*, consequently undermining the advantage of computational scaling during inference processes. To this end, inspired by the works [11, 12], we introduce a Kullback-Leibler (KL) divergence constraint, a popular and simple regularization technique to mitigate the homogenization output issue. To be specific, we aim to increase the reasoning output diversity across different steps, encouraging the model's multi-step reasoning process to gather multi-view insights, and better model the user's complex interest distribution, ultimately contributing to the overall sequence recommendation performance. Formally, we pair the predictive probability distributions of different reasoning states in pairwise combinations and maximize the KL divergence between these distribution pairs, which is equivalent to minimizing the following regularization term:

$$\mathcal{L}_{\text{KL}} = -\sum_{i=0}^{K-1}\sum_{j=i+1}^{K}\text{KL}(\hat{y}^{(i)}\|\hat{y}^{(j)}). \tag{6}$$

By combining the recommendation loss and the above KL regularization term, the overall learning objective for the ERL method is to minimize the following loss function:

$$\mathcal{L}_{\text{ERL}} = \mathcal{L}_{\text{Rec}} + \lambda\mathcal{L}_{\text{KL}}, \tag{7}$$

where λ is a hyperparameter that balances the constraint strength.

3.2.3 ***Inference Phase***. In the inference phase, we compute the inner product or cosine similarity (depending on the specific sequential recommendation algorithm) between user representation $\mathbf{h}_u$ and all candidate item representations, with top-scoring items selected as the final recommendation list.

3.3 Progressive Reasoning Learning (PRL)

Unlike the ensemble reasoning learning method, we explore another *Progressive Reasoning Learning (PRL)* mechanism. The core idea is to design a progressive distribution sharpening strategy to guide the intermediate reasoning chains, gradually approximating the user's true preference distribution. Intuitively, as the computational power allocated to the inference time increases, the recommendation model should be able to more accurately capture the fine-grained sequential features, narrowing the discrepancy between the predicted and actual user interest distribution.

3.3.1 ***Progressive Temperature Annealing (PTA)***. Drawing an analogy the human cognitive process, as the thinking depth increases, reasoning pathways become progressively refined until converging toward optimal solutions. Similarly, we expect that as the model's computations increases, the recommender would gradually clarify the user's interest evolving patterns, which is manifested as sharper predicted distributions. Inspired by this motivation, we propose a simple *Progressive Temperature Annealing (PTA)* method to guide the reasoning process. To achieve this, we first introduce a temperature coefficient, τ_k, for the k-th reasoning step to adjust the predicted distribution sharpness, which is formulated as follows:

$$\begin{aligned} \tau_k &= \tau * \alpha^{K-k}, \\ \hat{y}^{(k)} &= \text{softmax}(\mathbf{r}_k \cdot \mathbf{E}^\top / \tau_k), \end{aligned} \tag{8}$$

where τ is the base temperature, and α is a hyperparameter that controls the temperature decay rate.

In contrast to ensemble reasoning learning method, we apply separate recommendation losses to each reasoning hidden state to inject process supervision into the reasoning process, as follows:

$$\mathcal{L}_{\text{Rec}} = -\sum_{k=0}^{K} \log \hat{y}_{v^+}^{(k)}, \tag{9}$$

where $\hat{y}_{v^+}^{(k)}$ represents the logit corresponding to the v^+ item. With this lean annealing strategy, the model is encouraged to explore a broader solution space in the early reasoning stage, preventing it from getting stuck in local optima. Then, as the reasoning process progresses, the value of τ_k is gradually reduced to narrow the search space, guiding the model towards the global optimum. Thus, the proposed PTA can more effectively approximate the user's true preference distribution.

3.3.2 ***Reasoning-aware Contrastive Learning (RCL)***. However, relying solely on the temperature annealing strategy may not be sufficient to support the generalization ability of progressive reasoning learning. This is because, during the reasoning process, the model may suffer from the *reasoning bias*, where the model's reasoning direction deviates from the correct user interest distribution, ultimately leading to the accumulation of reasoning errors and deteriorating the reasoning capability. To address the above challenge, we design a novel *Reasoning-aware Contrastive Learning (RCL)* method to enhance the model's robust reasoning ability.

Specifically, we simulate the preceding accumulated reasoning error by injecting noise vectors into the reasoning states for each step, producing the noised reasoning input as follows:

$$\tilde{\mathbf{h}}_{n+i}^0 = \mathbf{h}_{n+i}^0 + \boldsymbol{\epsilon}, \quad i \in \{1, 2, \ldots, K\}, \tag{10}$$

where $\mathbf{h}_{n+i}^0$ is defined according to Eq. (2). The vector $\boldsymbol{\epsilon}$ represents the added noise embedding, sampled from a normal distribution, *i.e.*, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \gamma\mathbf{I})$, where $\mathbf{I} \in \mathbb{R}^d$ is the identity matrix of dimension d and γ controls the noise intensity. Then, we can obtain the new hidden state view $\tilde{\mathbf{R}} = [\tilde{\mathbf{r}}_1, \tilde{\mathbf{r}}_2, \ldots, \tilde{\mathbf{r}}_K]$ by feeding the noised input into the transformer encoder.

To enhance the model's robustness in reasoning denoising, we design a self-supervised task based on *Mutual Information Maximization (MIM)* [29, 31]. Formally, given variables X and Y, the *Mutual Information (MI)* measures the reduction in uncertainty of X after observing Y, which is defined as:

$$I(X, Y) = H(X) - H(X|Y),$$

where $H(\cdot)$ and $H(\cdot|\cdot)$ denote the entropy and conditional entropy of the random variable, respectively. By maximizing the MI between the original hidden states $\mathbf{R}$ and the denoised hidden states $\tilde{\mathbf{R}}$, it can effectively force the model to capture the essential sequential information from the user behavior data and historical reasoning process, achieving **self-reflection in the implicit thought space**.

However, directly maximizing mutual information is not feasible due to the intractability of the high-dimensional probability distribution estimation. Inspired by recent works [28, 34], we propose an InfoNCE-based reasoning contrastive learning method to optimize the lower bound of mutual information, which is defined as:

$$\mathcal{L}_{\text{RCL}} = -\sum_{k=1}^{K} \log \frac{\exp(\text{sim}(\tilde{\mathbf{r}}_k, \mathbf{r}_k^+)/\tau)}{\exp(\text{sim}(\tilde{\mathbf{r}}_k, \mathbf{r}_k^+)/\tau) + \sum_{\mathbf{r}_k^- \in \mathbf{R}_k^-} \exp(\text{sim}(\tilde{\mathbf{r}}_k, \mathbf{r}_k^-)/\tau)}, \tag{11}$$

where $\text{sim}(\cdot)$ denotes the dot product similarity function, $\mathbf{r}_k^+$ and $\mathbf{r}_k^-$ indicate the positive and negative contrastive hidden states at the k-th step, respectively. For the negative sample set $\mathbf{R}_k^-$, analogous to existing methods [26, 37], we utilize the k-th step reasoning states corresponding to the other item sequences within the same batch.

By combining the recommendation loss (*cf.* Eq. (9)) and the reasoning contrastive loss (*cf.* Eq. (11)), we can derive the overall objective function for the PRL method as follows:

$$\mathcal{L}_{\text{PRL}} = \mathcal{L}_{\text{Rec}} + \mathcal{L}_{\text{RCL}}. \tag{12}$$

3.3.3 ***Inference Phase***. During inference, we directly adopt the final reasoning step's output as the user representation, *i.e.*, $\mathbf{h}_u = \mathbf{r}_K$. Then, similar to Sec. 3.2.3, we compute similarity scores between $\mathbf{h}_u$ and the candidate item embedding matrix $\mathbf{E}$ to generate the recommendation list for the user u.

4 Experiments

In this section, we conduct extensive experiments and analyses to demonstrate the superiority of our proposed ReaRec framework.

Table 1: The statistics of experimental datasets.

	#Users	#Items	#Inter.	Sparisty
Yelp	13,083	10,697	443,807	99.68%
Video	89,021	22,933	530,989	99.97%
CDs	35,238	87,969	513,991	99.98%
Baby	140,292	30,689	780,809	99.98%

4.1 Experimental Setup

4.1.1 ***Datasets.*** To evaluate the effectiveness of our proposed methods, we conduct extensive experiments on five real-world recommendation datasets from Yelp and Amazon (Video & Games, CDs & Vinyl, and Baby Products) platforms. The detailed statistics of the datasets are summarized in Table 1.

4.1.2 ***Evaluation Metrics.*** We adopt top-k *Normalized Discounted Cumulative Gain (NDCG)* and top-k *Recall* to measure the recommendation performance, which are widely used in related sequential recommendation research [2, 25, 36]. In this paper, we specifically report **NDCG@{10,20}**, which assesses both the relevance and ranking quality of the top-k recommended items, and **Recall@{10,20}**, which evaluates the ability of the model to recall the ground-truth items in the top-k list.

4.1.3 ***Baselines.*** To thoroughly evaluate the generality of our proposed reasoning-enhanced framework, we conduct comprehensive benchmarking across different types of sequential recommendation models, including both ID-based and text-based encoding methods. The baselines are as follows: For ID-based encoding methods, we compare our methods with the following state-of-the-art models: **SASRec** [13] and **BERT4Rec** [24]. For Text-based encoding methods, we adopt **UniSRec** [10] and **MoRec** [38] as backbones.

4.2 Overall Performance

The recommendation performance of ID-based and text-based sequential models across all datasets is summarized in Table 2 and Table 3, respectively. We derive the following observations:

- For ID-based recommenders (*i.e.*, SASRec and BERT4Rec), we can find that BERT4Rec slightly outperforms SASRec at different metrics on most datasets. This suggests that incorporating both left and right contextual information enhances the model's ability to capture sequential patterns more effectively.
- Text-based methods (*i.e.*, UniSRec and MoRec) consistently outperform ID-based models across all datasets. For instance, on the Yelp dataset, UniSRec achieves a 9.51% improvement in NDCG@20 and a 14.14% increase in Recall@20 compared to SASRec. This improvement can be attributed to the ability of text-based models to leverage powerful language models for encoding item information, effectively mitigating data sparsity issues. In other words, by learning domain-invariant representations from textual feature spaces, these approaches effectively alleviate the recommendation bias, where underrepresented users and items are dominated by popular ones.
- Our proposed ERL and PRL methods, based on the ReaRec framework, consistently and significantly surpass baseline models at most cases. For example, for ID-based methods, ERL and PRL built on SASRec achieve average improvements of 6.76% and 8.21% respectively across all metrics on five datasets. Similarly, for text-based methods, ERL and PRL built on UniSRec outperform the base model by 12.29% and 10.43% on average. Unlike conventional SeqRec models, our reasoning-enhanced framework employs latent-space computations during the inference phase to deepen the feature crossing depth. This effectively unlock the latent reasoning power of various SeqRec backbones, demonstrating that increasing inference-time computation is a promising avenue for improving recommendation performance.

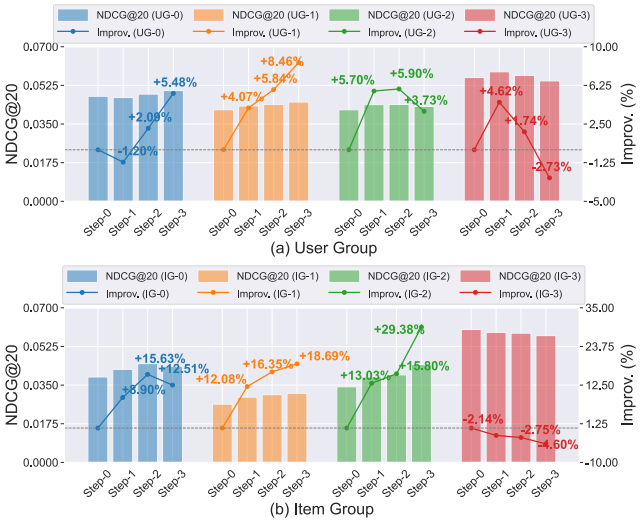

Figure 4: Robustness study w.r.t different user and item subgroups on Yelp dataset. 'Step-x' represents the recommendation performance at the x-th reasoning step. 'UG' and 'IG' denote User and Item Group, respectively, where higher group numbers indicate longer sequences and more popular items.

4.3 Further Analysis

4.3.1 ***Robustness Analysis Across User and Item Subgroups.*** To further analyze the robustness of our proposed ReaRec framework, we split users and items into different subgroups to gain deeper insights into the performance of the multi-step reasoning framework. Specifically, for users, we divide users into four equal-sized groups based on sequence length: {UG-0, UG-1, UG-2, UG-3}, where higher group numbers indicate longer sequences. For items, following previous work [26, 36], we group them into four groups based on interaction frequency: {IG-0, IG-1, IG-2, IG-3}, where higher group numbers indicate more popular items. We ensure each item group contains the same sample numbers. We fix the reasoning steps for PRL method during training at three, and analyze how recommendation performance changes for different user and item groups as reasoning steps increase during the inference phase. The detailed experimental results are shown in Fig. 4.

We can clearly observe distinct performance trends across different user and item subgroups. For short-sequence user groups and unpopular item groups, recommendation quality (NDCG@20)

Table 2: Performance comparison of different ID-based models on five datasets. 'N' and 'R' indicate NDCG and Recall metrics, respectively. 'Avg.' represents the average improvement rate across all metrics (*i.e.*, NDCG@{10,20} and Recall@{10,20}). Performance improvements are indicated by "↑", while performance declines are indicated by "↓".

Dataset	Method	SASRec					BERT4Rec				
		N@10	N@20	R@10	R@20	Avg.	N@10	N@20	R@10	R@20	Avg.
Yelp	Base	0.0347	0.0452	0.0626	0.1047	-	0.0364	0.046	0.0653	0.1038	-
	+ERL	0.0383	0.0474	0.0691	0.1056	**↑6.62%**	0.0371	0.0476	0.0661	0.1077	**↑2.60%**
	(Improv.)	(↑10.37%)	(↑4.87%)	(↑10.38%)	(↑0.86%)		(↑1.92%)	(↑3.48%)	(↑1.23%)	(↑3.76%)	
	+PRL	0.0388	0.0493	0.073	0.1149	**↑11.81%**	0.0377	0.0487	0.0708	0.1149	**↑7.14%**
	(Improv.)	(↑11.82%)	(↑9.07%)	(↑16.61%)	(↑9.74%)		(↑3.57%)	(↑5.87%)	(↑8.42%)	(↑10.69%)	
Video & Games	Base	0.0284	0.0353	0.0542	0.0816	-	0.0289	0.0355	0.0548	0.0810	-
	+ERL	0.0301	0.0385	0.0581	0.0915	**↑8.59%**	0.0311	0.0375	0.0578	0.0832	**↑5.36%**
	(Improv.)	(↑5.99%)	(↑9.07%)	(↑7.20%)	(↑12.13%)		(↑7.61%)	(↑5.63%)	(↑5.47%)	(↑2.72%)	
	+PRL	0.0299	0.0379	0.0572	0.0890	**↑6.81%**	0.0306	0.0380	0.0584	0.0879	**↑7.00%**
	(Improv.)	(↑5.28%)	(↑7.37%)	(↑5.54%)	(↑9.07%)		(↑5.88%)	(↑7.04%)	(↑6.57%)	(↑8.52%)	
CDs & Vinyl	Base	0.0148	0.0174	0.0317	0.0419	-	0.0149	0.0185	0.0326	0.0468	-
	+ERL	0.0182	0.0212	0.0363	0.0482	**↑18.59%**	0.0165	0.0208	0.0354	0.0524	**↑10.93%**
	(Improv.)	(↑22.97%)	(↑21.84%)	(↑14.51%)	(↑15.04%)		(↑10.74%)	(↑12.43%)	(↑8.59%)	(↑11.97%)	
	+PRL	0.0155	0.0195	0.0315	0.0470	**↑7.08%**	0.0162	0.0202	0.0334	0.0496	**↑6.59%**
	(Improv.)	(↑4.73%)	(↑12.07%)	(↓0.63%)	(↑12.17%)		(↑8.72%)	(↑9.19%)	(↑2.45%)	(↑5.98%)	
Baby Products	Base	0.0112	0.0157	0.0260	0.0437	-	0.0109	0.0154	0.0257	0.0439	-
	+ERL	0.0116	0.0164	0.0228	0.0418	↓2.16%	0.0148	0.0195	0.0293	0.0481	**↑21.49%**
	(Improv.)	(↑3.57%)	(↑4.46%)	(↓12.31%)	(↓4.35%)		(↑35.78%)	(↑26.62%)	(↑9.57%)	(↑14.01%)	
	+PRL	0.0135	0.0178	0.0281	0.0451	**↑11.30%**	0.0140	0.0185	0.0291	0.0466	**↑16.99%**
	(Improv.)	(↑20.54%)	(↑13.38%)	(↑8.08%)	(↑3.20%)		(↑28.44%)	(↑20.13%)	(↑6.15%)	(↑13.23%)	

Table 3: Performance comparison of different Text-based models on five datasets. 'N' and 'R' indicate NDCG and Recall metrics, respectively. 'Avg.' represents the average improvement rate across all metrics (*i.e.*, NDCG@{10,20} and Recall@{10,20}). Performance improvements are indicated by "↑", while performance declines are indicated by "↓".

Dataset	Method	UniSRec					MoRec				
		N@10	N@20	R@10	R@20	Avg.	N@10	N@20	R@10	R@20	Avg.
Yelp	Base	0.0380	0.0495	0.0737	0.1195	-	0.0391	0.0516	0.0757	0.1258	-
	+ERL	0.0406	0.0521	0.0770	0.1227	**↑4.81%**	0.0417	0.0531	0.0832	0.1283	**↑5.36%**
	(Improv.)	(↑6.84%)	(↑5.25%)	(↑4.48%)	(↑2.68%)		(↑6.65%)	(↑2.91%)	(↑9.91%)	(↑1.99%)	
	+PRL	0.0413	0.0529	0.0788	0.1253	**↑6.83%**	0.0410	0.0532	0.0804	0.1289	**↑4.16%**
	(Improv.)	(↑8.68%)	(↑6.87%)	(↑6.92%)	(↑4.85%)		(↑4.86%)	(↑3.10%)	(↑6.21%)	(↑2.46%)	
Video & Games	Base	0.0328	0.0421	0.0683	0.1054	-	0.0350	0.0438	0.0716	0.1065	-
	+ERL	0.0364	0.0440	0.0711	0.1015	**↑3.97%**	0.0392	0.0485	0.0744	0.1112	**↑7.76%**
	(Improv.)	(↑10.98%)	(↑4.51%)	(↑4.10%)	(↓3.70%)		(↑12.00%)	(↑10.73%)	(↑3.91%)	(↑4.41%)	
	+PRL	0.0352	0.0433	0.0658	0.0982	↓0.08%	0.0371	0.0462	0.0708	0.1067	**↑2.64%**
	(Improv.)	(↑7.32%)	(↑2.85%)	(↓3.66%)	(↓6.83%)		(↑6.00%)	(↑5.48%)	(↓1.12%)	(↑0.19%)	
CDs & Vinyl	Base	0.0150	0.0208	0.0298	0.0527	-	0.0186	0.0235	0.0405	0.0604	-
	+ERL	0.0208	0.0259	0.0428	0.0629	**↑31.54%**	0.0199	0.0248	0.0417	0.0609	**↑4.08%**
	(Improv.)	(↑38.67%)	(↑24.52%)	(↑43.62%)	(↑19.35%)		(↑6.99%)	(↑5.53%)	(↑2.96%)	(↑0.83%)	
	+PRL	0.0191	0.0253	0.0394	0.0640	**↑25.66%**	0.0198	0.0249	0.0417	0.0618	**↑4.42%**
	(Improv.)	(↑27.33%)	(↑21.63%)	(↑32.21%)	(↑21.44%)		(↑6.45%)	(↑5.96%)	(↑2.96%)	(↑2.32%)	
Baby Products	Base	0.0152	0.0199	0.0315	0.0501	-	0.0176	0.0231	0.0371	0.0588	-
	+ERL	0.0183	0.0239	0.0367	0.0589	**↑18.64%**	0.0184	0.0242	0.0373	0.0602	**↑3.06%**
	(Improv.)	(↑20.39%)	(↑20.10%)	(↑16.51%)	(↑17.56%)		(↑4.55%)	(↑4.76%)	(↑0.54%)	(↑2.38%)	
	+PRL	0.0182	0.0236	0.0359	0.0575	**↑16.77%**	0.0189	0.0247	0.0376	0.0611	**↑4.89%**
	(Improv.)	(↑19.74%)	(↑18.59%)	(↑13.97%)	(↑14.77%)		(↑7.39%)	(↑6.93%)	(↑1.35%)	(↑3.91%)	

Table 4: Inference time statistics for different steps. "Cost Inc." is short for Cost Increase, where higher values indicate greater time overhead. Note that the optimal performance typically corresponds to Step-2.

	Base	Step-1	Step-2	Step-3	Step-4	Step-5
SASRec	5.6761	5.7985	5.8752	5.9305	6.0310	6.2786
Cost Inc.	-	2.16%	3.51%	4.48%	6.25%	10.61%
BERT4Rec	5.6535	5.7685	5.9174	5.9621	6.0862	6.1224
Cost Inc.	-	2.03%	4.67%	5.46%	7.65%	8.29%
UniSRec	5.6061	5.6312	5.7596	5.8732	6.0303	6.0502
Cost Inc.	-	0.45%	2.74%	4.76%	7.57%	7.92%
MoRec	5.6638	5.7143	5.8391	5.9565	5.9659	5.9812
Cost Inc.	-	0.89%	3.10%	5.17%	5.33%	5.60%

Note: All time units are in second (s).

tends to steadily improve as the reasoning steps increase. For example, in the item group IG-1, more reasoning steps bring better performance gains of 12.08%, 16.35%, and 18.69%, respectively. In contrast, performance tends to decline for users with long interaction sequences and popular items as the reasoning steps increase. We speculate that this is primarily because longer user sequences provide richer contextual information, making it easier to mine interest evolution patterns. Beyond a certain point, additional inference computation fails to yield further performance improvements and even leads to performance degradation due to overthinking. Similarly, for high-popularity items, their well-trained representations allow the recommender to easily capture collaborative signals, making deeper feature crossing depth less beneficial. Overall, long-tail users and items usually require more thinking space to reason sparse interaction signals, whereas highly active users and items may not need redundant computational expansion. In the future, it may be necessary to develop differentiated fast and slow reasoning mechanism for different user sequences to further improve overall recommendation performance.

4.3.2 ***Impact of Reasoning Steps on Recommendation Performance.*** We investigate the variation trend of recommendation performance under different inference steps, that is, we train and perform inference using specified numbers of reasoning steps. We adopt NDCG@20 as the main evaluation metric. We compare the following approaches: **(1) Base**: The original SASRec sequential recommender serves as the baseline without reasoning enhancement; **(2) Naive**: Based on the Base method, we extend it to a multi-step reasoning paradigm, where the last hidden state is autoregressively fed back into the model, and only the final position is used directly as the user representation; **(3) RPE:** Building on the Naive approach, we further integrate *Reasoning Positional Embeddings* to bridge the task gap between sequence encoding mode and reasoning mode. Additionally, we also explore the performance of **(4) Ensemble Reasoning Learning (ERL)** and **(5) Progressive Reasoning Learning (PRL)** under multi-step reasoning.

As shown in Fig. 5, the Naive method, which lacks a specialized design, does not yield performance improvements and even underperforms compared to the base model. This is likely due to

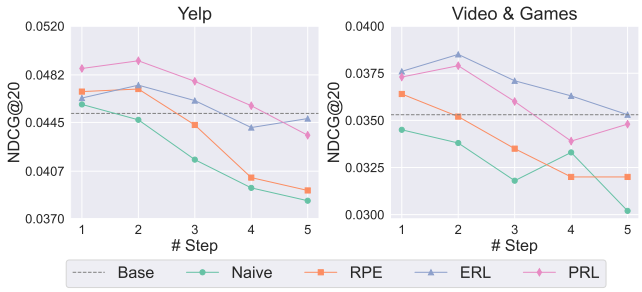

Figure 5: The performance variation trend of different methods under different reasoning steps.

the model's inability to distinguish between sequence encoding and the reasoning phases. Introducing reasoning positional embeddings (+RPE) effectively mitigates this task gap, yielding obvious performance gains. However, simply optimizing cross-entropy loss on the final-step output does not provide adequate supervision guidance for the intermediate reasoning states, potentially leading to reasoning pattern degradation and error accumulation. In contrast, our ERL and PRL methods significantly alleviate these issues by explicitly injecting stepwise supervision signals, reducing the optimization difficulty to some extent. Notably, as the number of inference steps increases, we observe a consistent performance decline across all methods. This suggests that excessive reasoning may trigger "**overthinking**"—simple user interaction patterns may not require intensive latent reasoning. Moreover, considering the post-hoc optimal step analysis in Fig. 2, developing an adaptive inference depth selection mechanism to balance reasoning depth and user sequence complexity presents a highly meaningful direction for future research.

4.3.3 ***Impact of Reasoning Steps on Inference Latency.*** Our ReaRec framework's expanded computational demands during inference introduce additional overhead. To evaluate this, we use the PRL method as an example, measuring the time cost on the test set as reasoning steps increase, as shown in Table 4. The results indicate that, despite adopting a recurrent autoregressive inference mechanism, the extra latency remains manageable. This efficiency stems from KV Caching technique, which significantly reduces attention computation complexity from $O(N^2)$ to $O(N)$ by reusing key and value vectors of past steps, thereby effectively minimizing redundant calculations. Further analysis with Fig. 5 reveals that our approaches generally achieve optimal performance at two reasoning steps. This means that our method increases performance by an average of 7.49% across all metrics with only a modest latency overhead of 3.51%, which is acceptable and practical for real-world deployment in industrial recommender systems. These results suggest that our efficient ReaRec framework holds great promise for real-world applications.

5 Conclusion

In this work, we pioneer the integration of deep reasoning into sequential recommendation by introducing **ReaRec**, a novel inference-time computing framework inspired by the *think-before-action* paradigm. Unlike traditional direct inference models, ReaRec expands

computational depth through multi-step implicit reasoning, enabling the SeqRec model to think before recommendation. We also propose two lightweight learning strategies to address the challenges of multi-step reasoning-process optimization: Ensemble Reasoning Learning (ERL) and Progressive Reasoning Learning (PRL), which enhance reasoning robustness and effectiveness. Extensive experiments across five real-world datasets validate the effectiveness and generalizability of our proposed ReaRec. Notably, ReaRec not only improves performance for long-tail users and items but also raises the performance ceiling of existing SeqRec backbones by up to 50% with post-hoc optimal step selection, highlighting the untapped potential of ReaRec for sequential recommendation. We believe our work opens a promising direction for future research at the intersection of reasoning and recommendation.

References

[1] Tesfaye Fenta Boka, Zhendong Niu, and Rama Bastola Neupane. 2024. A survey of sequential recommendation systems: Techniques, evaluation, and future directions. *Information Systems* (2024), 102427.

[2] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 108–116.

[3] Sunhao Dai, Ninglu Shao, Jieming Zhu, Xiao Zhang, Zhenhua Dong, Jun Xu, Quanyu Dai, and Ji-Rong Wen. 2024. Modeling user attention in music recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 761–774.

[4] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on ensemble learning. *Frontiers of Computer Science* 14 (2020), 241–258.

[5] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–42.

[6] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2023. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems* 36 (2023), 70757–70798.

[7] Binzong Geng, Zhaoxin Huan, Xiaolu Zhang, Yong He, Liang Zhang, Fajie Yuan, Jun Zhou, and Linjian Mo. 2024. Breaking the length barrier: Llm-enhanced CTR prediction in long textual user behaviors. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2311–2315.

[8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[9] Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, et al. 2024. DeepSeek-Coder: When the Large Language Model Meets Programming–The Rise of Code Intelligence. *arXiv preprint arXiv:2401.14196* (2024).

[10] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 585–593.

[11] HyeongJoo Hwang, Geon-Hyeong Kim, Seunghoon Hong, and Kee-Eung Kim. 2021. Multi-view representation learning via total correlation objective. *Advances in Neural Information Processing Systems* 34 (2021), 12194–12207.

[12] Yufei Jin, Heng Lian, Yi He, and Xingquan Zhu. 2024. HGDL: Heterogeneous Graph Label Distribution Learning. *Advances in Neural Information Processing Systems* 37 (2024), 40792–40830.

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

[14] Chenyi Lei, Yong Liu, Lingzi Zhang, Guoxin Wang, Haihong Tang, Houqiang Li, and Chunyan Miao. 2021. Semi: A sequential multi-modal information transfer network for e-commerce micro-video recommendations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3161–3171.

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

[16] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

[17] Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. 2024. Large language model empowered embedding generator for sequential recommendation. *arXiv preprint arXiv:2409.19925* (2024).

[18] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, et al. 2024. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592* 2 (2024).

[19] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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)*. Association for Computational Linguistics.

[20] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery* 8, 4 (2018), e1249.

[21] Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146* (2024).

[22] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).

[23] Uriel Singer, Haggai Roitman, Yotam Eshel, Alexander Nus, Ido Guy, Or Levi, Idan Hasson, and Eliyahu Kiperwasser. 2022. Sequential modeling with multiple attributes for watchlist recommendation in e-commerce. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 937–946.

[24] 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*. 1441–1450.

[25] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*. 598–606.

[26] Jiakai Tang, Sunhao Dai, Zexu Sun, Xu Chen, Jun Xu, Wenhui Yu, Lantao Hu, Peng Jiang, and Han Li. 2024. Towards Robust Recommendation via Decision Boundary-aware Graph Contrastive Learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2854–2865.

[27] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599* (2025).

[28] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? *Advances in neural information processing systems* 33 (2020), 6827–6839.

[29] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. 2019. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625* (2019).

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[31] Paul Viola and William M Wells III. 1997. Alignment by maximization of mutual information. *International journal of computer vision* 24, 2 (1997), 137–154.

[32] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830* (2019).

[33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[34] Junkang Wu, Jiawei Chen, Jiancan Wu, Wentao Shi, Xiang Wang, and Xiangnan He. 2023. Understanding contrastive learning via distributionally robust optimization. *Advances in Neural Information Processing Systems* 36 (2023), 23297–23320.

[35] Jian Xu, Sichun Luo, Xiangyu Chen, Haoming Huang, Hanxu Hou, and Linqi Song. 2025. RALLRec: Improving Retrieval Augmented Large Language Model Recommendation with Representation Learning. *arXiv preprint arXiv:2502.06101* (2025).

[36] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. 2023. Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2023*. 1063–1073.

[37] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1294–1303.

[38] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id- vs. modality-based recommender models revisited. In *Proceedings of the 46th*

International ACM SIGIR Conference on Research and Development in Information Retrieval. 2639–2649.

[39] Changshuo Zhang, Sirui Chen, Xiao Zhang, Sunhao Dai, Weijie Yu, and Jun Xu. 2024. UOEP: User-Oriented Exploration Policy for Enhancing Long-Term User Experiences in Recommender Systems. *arXiv preprint arXiv:2401.09034* (2024).

[40] Kepu Zhang, Teng Shi, Sunhao Dai, Xiao Zhang, Yinfeng Li, Jing Lu, Xiaoxue Zang, Yang Song, and Jun Xu. 2024. SAQRec: Aligning Recommender Systems to User Satisfaction via Questionnaire Feedback. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 3165–3175.

[41] Xiao Zhang, Sunhao Dai, Jun Xu, Zhenhua Dong, Quanyu Dai, and Ji-Rong Wen. 2022. Counteracting user attention bias in music streaming recommendation via reward modification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2504–2514.

[42] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.