Calibration-Disentangled Learning and Relevance-Prioritized Reranking for Calibrated Sequential Recommendation

Hyunsik Jeon
University of California, San Diego
San Diego, CA, USA
hyjeon@ucsd.edu

Se-eun Yoon
University of California, San Diego
San Diego, CA, USA
seeuny@ucsd.edu

Julian McAuley
University of California, San Diego
San Diego, CA, USA
jmcauley@ucsd.edu

ABSTRACT

Calibrated recommendation, which aims to maintain personalized proportions of categories within recommendations, is crucial in practical scenarios since it enhances user satisfaction by reflecting diverse interests. However, achieving calibration in a sequential setting (i.e., calibrated sequential recommendation) is challenging due to the need to adapt to users' evolving preferences. Previous methods typically leverage reranking algorithms to calibrate recommendations after training a model without considering the effect of calibration and do not effectively tackle the conflict between relevance and calibration during the reranking process. In this work, we propose LeapRec (Calibration-Disentangled Learning and Relevance-Prioritized Reranking), a novel approach for the calibrated sequential recommendation that addresses these challenges. LeapRec consists of two phases, model training phase and reranking phase. In the training phase, a backbone model is trained using our proposed calibration-disentangled learning-to-rank loss, which optimizes personalized rankings while integrating calibration considerations. In the reranking phase, relevant items are prioritized at the top of the list, with items needed for calibration following later to address potential conflicts between relevance and calibration. Through extensive experiments on four real-world datasets, we show that LeapRec consistently outperforms previous methods in the calibrated sequential recommendation. Our code is available at https://github.com/jeon185/LeapRec.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

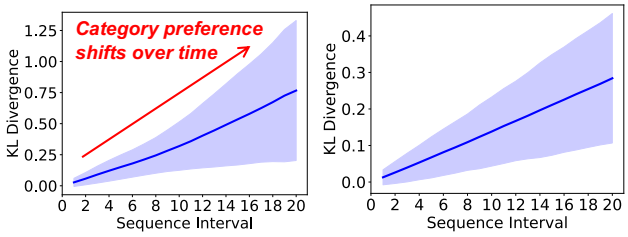
calibrated sequential recommendation; calibration-disentangled learning; relevance-prioritized reranking

ACM Reference Format:

Hyunsik Jeon, Se-eun Yoon, and Julian McAuley. 2024. Calibration-Disentangled Learning and Relevance-Prioritized Reranking for Calibrated Sequential Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3627673.3679728

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679728



(a) ML-1M (b) Steam

Figure 1: KL divergence analysis of user-interacted category distributions over sequence intervals, employing a window size of 20 for each category distribution. This plot illustrates the shifts in category preferences over time in real-world datasets: ML-1M and Steam. Detailed data statistics are summarized in Table 1.

1 INTRODUCTION

Calibrated recommendation aims to reflect a user's diverse interests within a recommendation list by maintaining the proportions of various categories observed in past interactions [1, 3, 38, 39]. For instance, if a user has historically watched 70% drama and 30% action movies, calibrated recommendation should suggest a list of movies maintaining a similar genre ratio. This problem differs from studies like [27, 44, 46] that define calibration in terms of probability-based user preference estimation, such as estimating how likely a user will prefer an item. To achieve the calibrated recommendation, two potentially conflicting objectives must be addressed: 1) *relevance*, which aligns with the user's current preferences, and 2) *calibration*, which sustains consistency with their long-term category interests. This challenge becomes particularly significant in sequential settings (i.e., calibrated sequential recommendation) where users' category preferences shift over time. These dynamic shifts in preferences are depicted in Figure 1, where the Kullback–Leibler (KL) divergence between category distributions increases as the sequence interval extends, highlighting the intricate challenge of balancing relevance with calibration.

Existing work on calibrated recommendation focuses on post-processing approaches [1, 38, 39]. Specifically, a recommendation model is first trained to meet the relevance objective; then, the model output is reranked to meet the calibration objective. The difference between these methods lies in reranking, such as greedy (CaliRec [39]), mixed integer programming (MIP [38]), and minimum-cost flow (MCF [1]) algorithms. However, applying calibration during reranking can lead to degradation of accuracy because they do not consider the impacts of calibration during the training phase.

Hence, these methods encounter challenges in maintaining the accuracy of recommendations in the reranking phase, since the changes required for calibration can conflict with the relevance-based ranking criteria, especially in sequential settings where category preference shifts over time as shown in Figure 1. Recently, DACSR [3] introduced an end-to-end model designed to optimize both relevance and calibration during the training phase. However, DACSR optimizes the calibration across the entire item set to ensure the loss function is differentiable although the primary goal is to calibrate the top-k recommendations. As a result, DACSR struggles to simultaneously optimize for both accuracy and calibration, particularly for the top-k recommendations.

In this work, we propose LeapRec (Calibration-Disentangled Learning and Relevance-Prioritized Reranking), a method that considerably improves upon previous work by novel training and reranking schemes. First, LeapRec learns to disentangle calibration from relevance during the training phase by optimizing our proposed *calibration-disentangled learning-to-rank* loss. This approach enables the model to learn rankings adaptable to changes in calibration, allowing for accurate ranking even when calibration is extensively considered. Consequently, even when calibration adjustments are applied during reranking, the calibration is effectively enhanced without sacrificing much accuracy. After the training phase, LeapRec further optimizes for relevance and calibration through our proposed *relevance-prioritized reranking*. This approach prioritizes placing relevant items at the top of the recommendation list while maintaining calibration across the entire list. This strategy effectively mitigates the risk of excluding items from categories previously unexplored by users, a limitation encountered with previous calibration objectives.

Our contributions are as follows:

- We propose a novel learning method to effectively disentangle calibration from relevance in the training phase, thus enabling us to facilitate adaptable ranking under varying calibration considerations.
- We propose a novel reranking scheme that effectively enhances relevance and calibration by prioritizing relevant items while ensuring consistent calibration across the list.
- We conduct extensive experiments on four real-world datasets and show that LeapRec achieves superior performance compared to existing approaches.

2 PRELIMINARIES

2.1 Problem Definition

The problem of calibrated sequential recommendation is defined as follows. Let $\mathcal{U}$, $\mathcal{I}$, and C be the sets of users, items, and categories, respectively. Each item $i \in \mathcal{I}$ is associated with a set of categories $C^i = \{c_1^i, c_2^i, ..., c_N^i\}$, where each $c_n^i \in C$ represents a category of item i, and N is the number of categories, varying per item. For each user $u \in \mathcal{U}$, we have sequential interactions $\mathcal{S}^u = (s_1^u, s_2^u, \ldots, s_T^u)$, where $s_t^u \in \mathcal{I}$ is user u's interacted item at step t, and T denotes the length of the sequence which varies among users. Given sequences $\mathcal{S}^u$ of all users $u \in \mathcal{U}$, our goal is to recommend each user an item list $\mathcal{R}^u = (r_1^u, r_2^u, \ldots, r_K^u)$ that is relevant to the user's future needs (i.e., accurate), while reflecting the user's sequential category interests with their appropriate proportions (i.e., sequentially calibrated), at step $T+1$; $r_k^u \in \mathcal{I}$ is the k'th recommended item to user u where lower k indicates higher rank. The metrics for measuring the degree of calibration are detailed in Section 2.2.

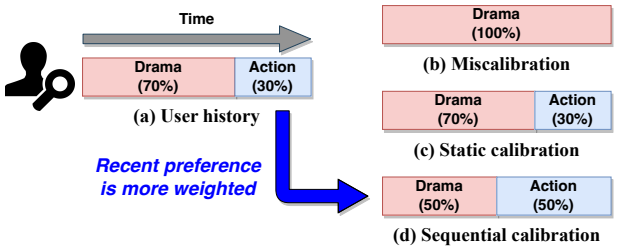


Figure 2: Given user history (a), sequential calibration (d) applies more weight to recent category preference, in contrast to recommendation without calibration (b) and static calibration (c).

The primary challenge in calibrated sequential recommendation lies in adeptly balancing the often conflicting demands of relevance, which reflects a user's immediate preferences, and calibration, which ensures consistency with their long-term category interests. This task is further complicated by the constantly evolving nature of user preferences.

2.2 Calibration Metrics for Sequential Recommendation

The degree of calibration is measured by comparing the category distributions of items in a user's past interactions and items in their recommended list [39]. Specifically, it is defined as the divergence between these two distributions (i.e., miscalibration), where a lower value indicates superior calibration performance. Steck [39] proposed metrics for miscalibration under various criteria such as whether the user interactions are treated as equally or weighted regarding their recency, as illustrated in Figure 2. In this work, we adopt *sequential miscalibration* as our calibration metric, specifically tailored for sequential recommendations. This metric is described in the following definition.

Definition 1 (Sequential miscalibration)**:** *Given user u's sequential interactions $\mathcal{S}^u$ and the user's recommendation list $\mathcal{R}^u$, the sequential miscalibration $\mathcal{S}_{KL}(u)$ is defined as follows:*

$$\mathcal{S}_{KL}(u) = KL(p \| \tilde{q}) = \sum_{c \in C} p(c|u) \log \frac{p(c|u)}{\tilde{q}(c|u)}, \tag{1}$$

where

$$p(c|u) = \frac{\sum_{s_t^u \in \mathcal{S}^u} \alpha^{T-t} \cdot p(c|s_t^u)}{\sum_{s_t^u \in \mathcal{S}^u} \alpha^{T-t}}, \tag{2}$$

$$q(c|u) = \frac{\sum_{r_k^u \in \mathcal{R}^u} p(c|r_k^u)}{|\mathcal{R}^u|}, \tag{3}$$

$$\tilde{q}(c|u) = (1-\beta) q(c|u) + \beta p(c|u), \tag{4}$$

$KL(\cdot)$ indicates the Kullback–Leibler (KL) divergence between two distributions, $T = |\mathcal{S}^u|$, and $\alpha, \beta \in (0, 1)$ are hyperparameters.

In Equations (2) and (3), $p(c|s_t^u)$ and $p(c|r_k^u)$ are the category distributions of items $s_t^u, r_k^u \in \mathcal{I}$, respectively. If an item is associated with multiple categories, each category is equally weighted in the

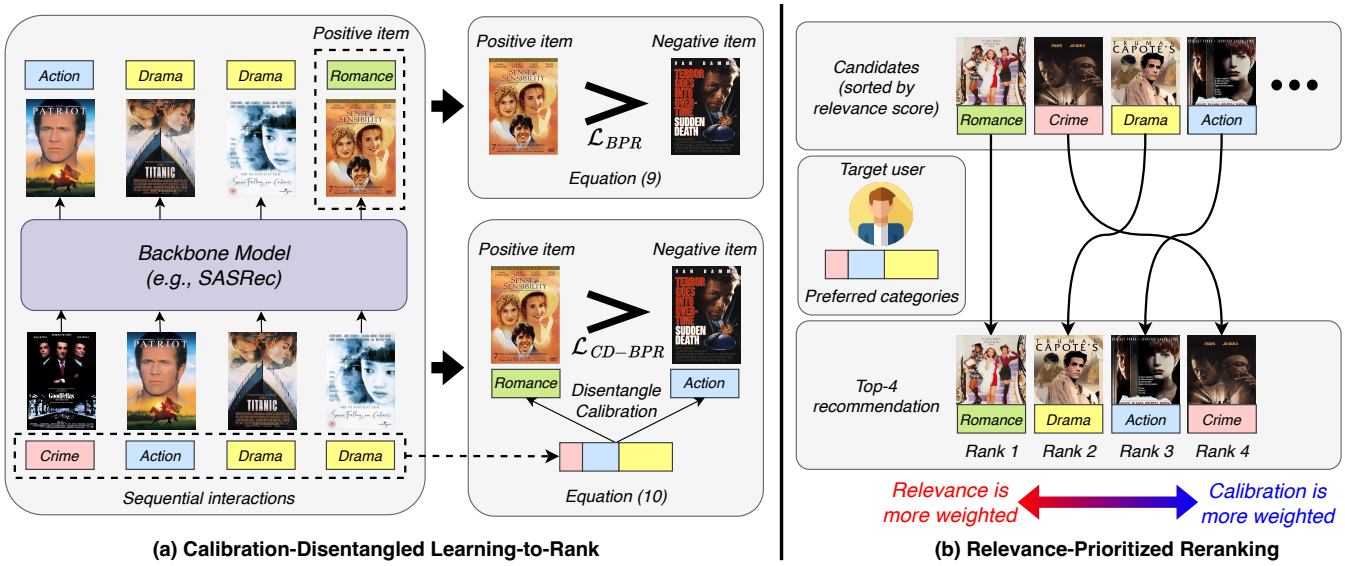


Figure 3: LeapRec consists of two phases: (a) calibration-disentangled learning-to-rank and (b) relevance-prioritized reranking. In the first phase, a backbone model is trained to optimize personalized rankings, accommodating both with and without calibration considerations. In the second phase, items are greedily added to the recommendation list, where relevance is prioritized at higher ranks and calibration at lower ranks.

distribution. In Equation (2), α (e.g., 0.9) enables us to consider recent interests more weighted to the calibration, thus differentiating this metric from static miscalibration. In Equation (4), we modify $q(c|u)$ to $\tilde{q}(c|u) = (1-\beta)q(c|u) + \beta p(c|u)$ with a small value of β (e.g., 0.01), ensuring the KL divergence remains well-defined and does not diverge, as in previous work [39].

3 PROPOSED METHOD

In this section, we describe LeapRec (Calibration-Disentangled Learning and Relevance-Prioritized Reranking), a novel approach for calibrated sequential recommendation. Figure 3 depicts the overall process of LeapRec, which consists of two phases: model training phase and reranking phase. During the training phase, LeapRec employs a sequential recommendation model, such as SASRec [20], as its backbone. It optimizes our proposed *calibration-aware learning-to-rank* loss, which is designed to disentangle calibration from relevance, enabling the model to estimate accurate personalized rankings regardless of whether calibration is considered. During the reranking phase, LeapRec applies our proposed *relevance-prioritized reranking* algorithm. This algorithm adjusts the model's output for each user to reduce miscalibration, while prioritizing the most relevant items in the final recommendations.

3.1 Calibration-Disentangled Learning-to-Rank

The objective of the training phase is to train a sequential model that can predict the probability a user will interact with an item based on their past interactions. Formally, given a user u's sequence of interactions $\mathcal{S}^u = (s_1^u, s_2^u, \ldots, s_T^u)$ where $s_t^u \in \mathcal{I}$ is the interacted item at step t, the model aims to estimate the probability for all items $v \in \mathcal{I}$ at step $T+1$:

$$p(s_{T+1}^u = v | \mathcal{S}^u). \tag{5}$$

Generally, sequential models are trained to increase the gap between the relevance scores of interacted items and non-interacted items using pointwise [15], pairwise [35] or setwise [31] losses. Various frameworks, including Markov Chains [10, 36], Recurrent Neural Networks (RNN) [13, 23, 24, 28], Convolutional Neural Networks (CNN) [41, 50], and self-attention mechanisms [20, 29, 40], are effectively used in these sequential models, demonstrating high performance in terms of accuracy.

Let $f_\theta(u, i, t)$ represent the relevance score between user u and item i at step t, with parameters θ. Existing calibrated recommendation methods [1, 38, 39] often rely solely on post-processing techniques and thus overlook the potential impact of calibration adjustments on the final ranking order. However, it is crucial to integrate calibration directly into the training process to anticipate how rankings might change when calibration is applied. This proactive approach is essential for achieving high performance in both accuracy and calibration. To illustrate the importance of integrating calibration directly into the training process, consider a scenario where relevance scores indicate a preference for item i over item j based on past interactions (i.e., $f_\theta(u, i, t) > f_\theta(u, j, t)$). After the training phase, calibration scores may compel the system to rank item j higher than item i. It is difficult to determine whether these items should be reranked, since the degree to which item i is more relevant than item j becomes uncertain when calibration is taken into account. Thus, ensuring that the model can maintain consistent rankings even after calibration adjustments during reranking is crucial for the calibrated sequential recommendation.

To address this issue, we propose a calibration-disentangled learning-to-rank, a model-agnostic learning approach. For brevity, let $f_\theta(u, i, t) := r_{i,t}^u$. Suppose user u interacted with item i instead of item j at step t. This interaction indicates that the user prefers item i over item j, even with category preference taken into account. Thus, at the recommendation step t, it is crucial to recommend item

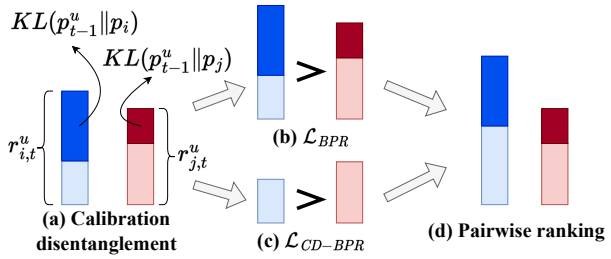


Figure 4: Illustrative example of calibration-disentangled learning-to-rank. Initially, miscalibration scores are disentangled from relevance scores (a). Then, we train the model based on both $\mathcal{L}_{BPR}$ (b) and $\mathcal{L}_{CD-BPR}$ (c), resulting in a personalized pairwise ranking considering calibration (d).

i to user u over item j, even subsequent to calibration. Existing methods learn a preference order without considering the category preference as follows:

$$r^u_{i,t} > r^u_{j,t}. \tag{6}$$

This order does not guarantee that item i will be prioritized over item j after considering calibration. Hence, we propose to disentangle calibration from pairwise ranking, which learns that item i is preferred over item j even with calibration considerations:

$$r^u_{i,t} - KL(p^u_{t-1}||p_i) > r^u_{j,t} - KL(p^u_{t-1}||p_j), \tag{7}$$

where p^u_{t-1} is the sequential category preference of user u at step $t-1$, and p_i and p_j are the category distributions of items i and j, respectively. $KL(\cdot)$ is the miscalibration score computed by Equation (1). By integrating calibration directly into the training phase, our method trains the model to optimize relevance while being aware of calibration needs. This ensures that adjustments made for calibration in the reranking phase do not negatively impact the relevance of the recommendations, thereby maintaining accuracy when enhancing calibration. To learn these two pairwise ranking losses (Equations (6) and (7)), we propose the *calibration-disentangled learning-to-rank* loss, which extends the Bayesian Personalized Ranking (BPR) loss as follows:

$$\mathcal{L} = \mathcal{L}_{BPR} + \gamma \mathcal{L}_{CD-BPR}, \tag{8}$$

where

$$\mathcal{L}_{BPR} = \sum_{u \in \mathcal{U}} \sum_{t \in [1,\dots,T]} -\log \sigma \left(r^u_{i,t} - r^u_{j,t} \right), \tag{9}$$

$$\mathcal{L}_{CD-BPR} = \sum_{u \in \mathcal{U}} \sum_{t \in [1,\dots,T]} -\log \sigma \left(r^u_{i,t} - KL(p^u_{t-1}||p_i) - r^u_{j,t} + KL(p^u_{t-1}||p_j) \right), \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function and $\gamma \in \mathbb{R}$ is a hyperparameter that determines the importance of $\mathcal{L}_{CD-BPR}$. The calibration-disentangled learning-to-rank does not require learning additional parameters, thus avoiding an increase in model complexity.

The key aspect of calibration-disentangled learning-to-rank is its ability to dynamically adjust relevance scores by incorporating calibration scores based on the user's category preferences. As illustrated in Figure 4, this method initially separates the miscalibration scores from relevance scores (Figure 4 (a)). For instance, if the categories of item i significantly diverge from user u's preferences before t, the model f_θ is prompted to increase the relevance score

Algorithm 1 Overall process of LeapRec

Input: $\{\mathcal{S}^u : u \in \mathcal{U}\}$ and hyperparameters (α, β, γ, and λ)
Output: $\{\mathcal{R}^u : u \in \mathcal{U}\}$

while stop condition is not met **do** ▷ Train a backbone
  for $u \in \mathcal{U}$ **do**
    Optimize f_θ to minimize $\mathcal{L}$ ▷ Equation (8)
  end for
end while
for $u \in \mathcal{U}$ **do** ▷ Rerank
  $C^u = \{\}$
  for $k \in [1, K]$ **do**
    $i \leftarrow \max_{i \in \mathcal{I} \setminus C^u} (1-\lambda^{1/k}) r^u_{i,T+1} - \lambda^{1/k} \Delta \mathcal{S}_{KL}(i|C^u)$
    $C^u \leftarrow C^u \cup \{i\}$
  end for
  $\mathcal{R}^u \leftarrow C^u$
end for

$r^u_{i,t}$ (Figure 4 (d)) to compensate for the high miscalibration score $KL(p^u_{t-1}|p_i)$, using $\mathcal{L}_{CD-BPR}$ (Figure 4 (c)). This approach differs from previous methods that rely solely on $\mathcal{L}_{BPR}$, where personalized ranking might change unpredictably after calibration during reranking. However, relying only on $\mathcal{L}_{CD-BPR}$ also can lead to issues in personalized rankings. For example, consider a case where $r^u_{i,t} - KL(p^u_{t-1}||p_i)$ is larger than $r^u_{j,t} - KL(p^u_{t-1}||p_j)$, but $r^u_{i,t}$ is lower than $r^u_{j,t}$. In such case, $\mathcal{L}_{CD-BPR}$ prioritizes item i over item j when calibration is considered, but it may overlook genuine relevance if calibration is not considered. This case verifies the necessity of integrating both $\mathcal{L}_{BPR}$ and $\mathcal{L}_{CD-BPR}$.

3.2 Relevance-Prioritized Reranking

The reranking phase aims to maximize both accuracy and calibration using the trained model f_θ. The goal is to recommend user u a list of items $\mathcal{R}^u = (r^u_1, r^u_2, \dots, r^u_K)$ where $r^u_k \in \mathcal{I}$ is the k'th recommended item. Reranking generates the list by selecting the most suitable items among the candidates. The main challenge in the reranking phase is to measure which item is the best for the user at each step, considering both accuracy and calibration.

From Figure 1, we observe that a user is likely to interact with an item that is associated with a category the user has not preferred before. For instance, a user who predominantly watches action movies might develop an interest in romance movies due to temporal factors. In this case, it is necessary to ensure that items that users like are recommended regardless of their category to satisfy the user's future needs. However, naively using weighted sum [1, 38, 39] may not adequately handle such cases. In the example, the romance movie may have a low overall score despite its high relevance due to the high miscalibration score, since it contrasts with the user's past category preferences. This conflict of relevance and calibration should be treated as a critical issue in the calibrated sequential recommendation, yet it has not been thoroughly addressed in previous works [1, 3, 38, 39].

To address this challenge, our reranking strategy prioritizes a user's emerging interests by integrating both relevance and calibration but favoring relevance in the higher ranks of the recommendation list. If we consider that the backbone model f_θ is trained to predict the user's evolving preferences based on a sequential model,

Table 1: Summary of four real-world datasets used in this work.

Dataset	# Users	# Items	# Categories	# Interactions	Avg. sequence len.	User-item density	Avg. # categories
ML-1M	6,038	3,883	18	575,281	95.28	0.0245	1.6503
Goodreads	16,765	25,474	10	954,958	56.96	0.0022	3.6269
Grocery	54,882	39,853	26	438,681	7.99	0.0002	1.0000
Steam	242,223	14,419	22	2,732,749	11.29	0.0008	2.6242

we can infer that the relevance scores from the model are more closely related to a user's emerging interests. Thus, we propose the *relevance priority* property for the reranking algorithm as follows:

Property 1 (Relevance priority)**:** *In higher-ranked recommendations, relevance should be prioritized over calibration.*

Reranking based on this property prevents potentially relevant items from being lower ranked (or excluded) due to calibration constraints, thereby offering a more accurate reflection of the user's evolving interests. Such prioritized approach has recently been explored in previous work on multi-objective recommendation, demonstrating its effectiveness [18].

To apply Property 1 in the reranking algorithm, we propose a simple yet effective objective function for each user u as follows:

$$\max_{\mathcal{R}^u, |\mathcal{R}^u|=K} \left((1-\lambda^{1/k}) \sum_{i \in \mathcal{R}^u} r^u_{i,T+1} - \lambda^{1/k} \mathcal{S}_{KL}(u) \right), \tag{11}$$

where $\lambda \in [0,1]$ is a balancing hyperparameter between relevance and calibration, $k \in [1,K]$ indicates the position in the recommendation list, $r^u_{i,T+1}$ is the relevance score of item i for user u at step $T+1$ (i.e., $f_\theta(u,i,T+1)$), and $\mathcal{S}_{KL}(u)$ is sequential miscalibration which is defined in Equation (1). Smaller k (i.e., higher ranking) assigns more weight to relevance and larger k (i.e., lower ranking) to calibration. Hence, the objective function satisfies the relevance priority property. Moreover, we leverage sequential miscalibration $\mathcal{S}_{KL}(u)$ rather than static miscalibration used in most previous methods [1, 3, 38] to consider the recent category preferences in measuring the degree of calibration.

Finding the optimal recommendation list from Equation (11) is a combinatorial optimization problem and NP-hard. Thus, we adopt a greedy approach, which is fast and effective, to optimize the objective function. Specifically, at the k'th recommendation for user u, we select an item that maximizes the gain of scores among items that are yet to be selected as follows:

$$\max_{i \in \mathcal{I} \setminus C^u} \left((1-\lambda^{1/k}) r^u_{i,T+1} - \lambda^{1/k} \Delta \mathcal{S}_{KL}(i|C^u) \right), \tag{12}$$

where $C^u \subseteq \mathcal{R}^u$ is the current recommendation list for user u, and $\Delta\mathcal{S}_{KL}(i|C^u)$ indicates the difference of $\mathcal{S}_{KL}(u)$ when item i is added to the current recommendation list C^u.

3.3 Overall Process of LeapRec

The overall process of LeapRec consists of the backbone model training phase and the reranking phase. Algorithm 1 shows how LeapRec trains the backbone model and reranks the results. Given users' sequential interactions $\{\mathcal{S}^u : u \in \mathcal{U}\}$, LeapRec returns recommendation lists $\{\mathcal{R}^u : u \in \mathcal{U}\}$ for all users. In lines 1 to 5, LeapRec trains a backbone model by minimizing the loss in Equation (8) for predefined epochs (e.g., 100); in line 2, we adopt a mini-batch training in our practical implementation. Then, in lines 6 to 13, LeapRec greedily selects K items for each user considering both relevance and sequential miscalibration.

4 EXPERIMENTS

In this section, we conduct experiments to answer the following questions.

- Q1. **Performance comparison (Section 4.2).** Does LeapRec provide better trade-off between accuracy and calibration compared to competitors? How efficient and fast is LeapRec compared to competitors in generating recommendations?
- Q2. **Ablation study (Section 4.3).** Do the main components in LeapRec help improve the performance?
- Q3. **Effect of the balancing hyperparameter (Section 4.4).** How does the balancing hyperparameter λ, which is the key factor in enhancing calibration, affect the overall recommendation?
- Q4. **Case study (Section 4.5).** How LeapRec recommend a list of items considering both relevance and calibration?

4.1 Experimental Setting

4.1.1 Datasets. We evaluate LeapRec and other methods using four real-world datasets from distinct domains: movies (ML-1M[1] [8]), books (Goodreads[2] [43]), grocery products (Grocery[3] [11, 32]), and video games (Steam[4] [34]). These datasets are chosen for their diversity in domain, sparsity level, and number of categories, as detailed in Table 1. In ML-1M dataset, we follow previous work [1, 39] and consider only ratings of four stars and above, simulating positive feedback.

4.1.2 Backbone Model. Unless otherwise stated, we consider SASRec [20] as a backbone framework which has shown its superior performance compared with other frameworks in comprehensive experiments [20, 22]. However, the calibration-disentangled learning-to-rank approach is open to other sequential recommendation frameworks such as GRU4Rec [13], Caser [41], and BERT4Rec [40] since it is a model-agnostic learning approach.

4.1.3 Baseline Methods. We compare LeapRec with the following four existing calibration recommendation methods.

- **CaliRec** [39] is a post-processing approach that reranks the output of a backbone model using a greedy algorithm, optimizing for static calibration.
- **CaliRec$^+$** [39] differs from CaliRec in reranking algorithm where it optimizes for sequential calibration instead of static calibration. We adopt $\mathcal{S}_{KL}(u)$ as in LeapRec for the miscalibration score.
- **MIP** [38] is a post-processing approach that utilizes mixed integer programming, focusing on achieving static calibration.

[1] https://grouplens.org/datasets/movielens/1m
[2] https://cseweb.ucsd.edu/~jmcauley/datasets.html#goodreads
[3] https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews
[4] https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data

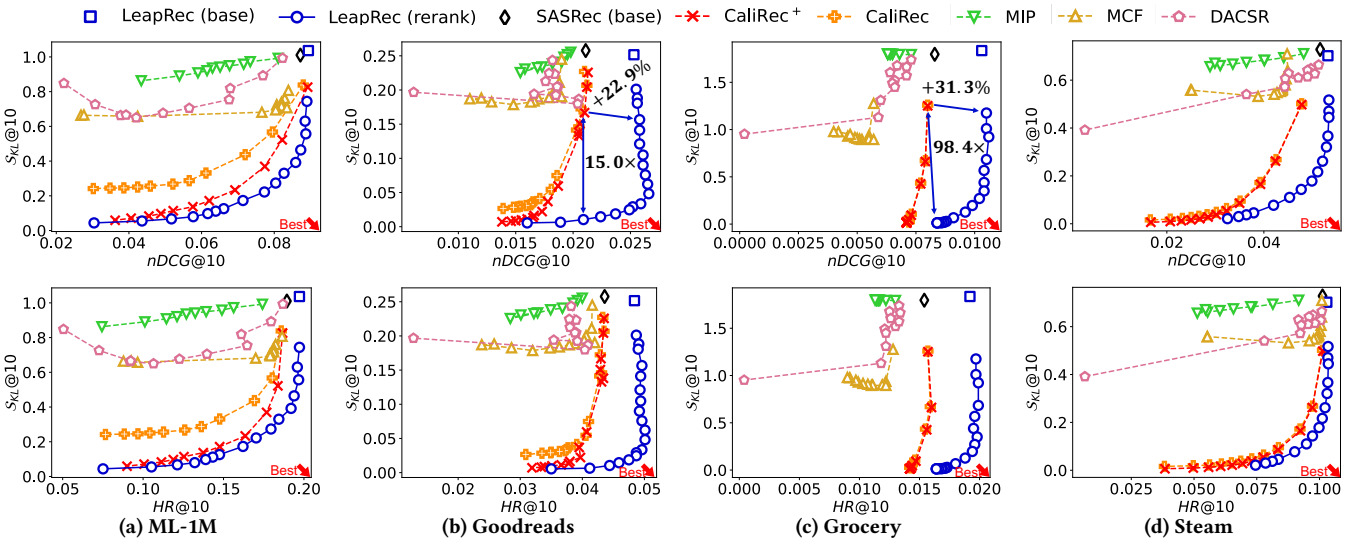


Figure 5: Trade-off comparison of LeapRec and baselines. The first row shows $nDCG@10$ vs. $\mathcal{S}_{KL}@10$, and the second row shows $HR@10$ vs. $\mathcal{S}_{KL}@10$. LeapRec outperforms the baselines on four real-world datasets, drawing better trade-off curves between accuracy and calibration.

Table 2: Comparative analysis of computational complexity for reranking algorithms between LeapRec and baselines. Variables include n (number of users), m (number of items), k (recommendation list size), and c (number of categories).

Method	LeapRec (ours) CaliRec CaliRec+	MIP	MCF
Complexity	$O(nkm)$	$O(nm^k)$	$O(n(k+c)m^2 \log m)$

- **MCF** [1] is a post-processiong approach that employs a minimum-cost flow algorithm in its reranking phase to adjust for static calibration.
- **DACSR** [3] is an end-to-end approach that simultaneously targets accuracy and static calibration optimization.

Each method uses hyperparameter λ to balance between accuracy and calibration.

4.1.4 Experimental Process. We follow the *leave-one-out* protocol as established by prior studies [17, 18, 21, 40, 45]. For each user u, we split their historical interaction sequence $\mathcal{S}^u$ into three parts: the most recent interaction for testing, the second most recent one for validation, and all earlier interactions for training. For LeapRec and the baseline methods (CaliRec, MIP, MCF, and DACSR), we conduct training of models for 200 epochs on ML-1M dataset (100 epochs on the other datasets) and choose the models with the best validation performance. Subsequently, we apply each method's specific reranking strategy (except for DACSR), adjusting the balancing hyperparameter λ to draw trade-off curves between relevance and calibration.

4.1.5 Evaluation Metrics. We evaluate the performance in two criteria accuracy and calibration, by investigating the trade-off curve between them. We use hit ratio (HR@K) and normalized discounted cumulative gain (nDCG@K) metrics to measure the accuracy. Given top-K recommendation lists, HR@K measures whether the lists contain the ground-truth items, and nDCG@K weighs the rank of ground-truth items in the list. We use sequential miscalibration ($S_{KL}@K$) metrics to measure the calibration (see Section 2.2 for details). Higher HR@K and nDCG@K indicate better performance, while lower $S_{KL}@K$ indicates better performance. We set K to 10 in our experiments.

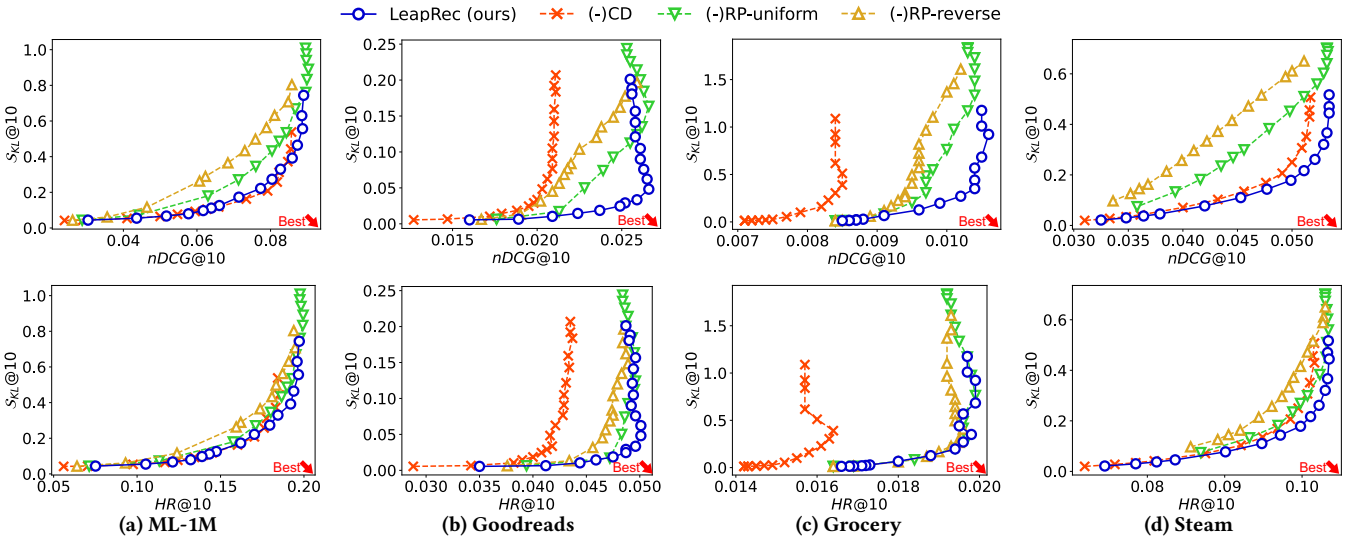
Table 3: Comparison of running times (in seconds) for reranking algorithms between LeapRec and baselines across various datasets. Bold indicates the fastest record in each row.

Method	LeapRec (ours)	CaliRec	CaliRec+	MIP	MCF
ML-1M	**24**	**24**	**24**	1,793	330
Goodreads	232	**228**	229	14,018	1,076
Grocery	2,808	2,816	**2,799**	18,176	3,015
Steam	**3,925**	4,121	3,944	92,245	16,256

4.1.6 Hyperparameters. We implement LeapRec using PyTorch [33]. For all datasets, we set α (Equation (2)), β (Equation (4)), and γ (Equation (8)) to 0.9, 0.01, and 0.1, respectively. For a backbone SASRec [20], we set the learning rate, the dimension of embedding (e.g., user embedding and item embedding), batch size, and the number of self-attention blocks to 0.001, 50, 128, and 2, respectively. Moreover, the dropout rate is set to 0.2 for ML-1M and 0.5 for the other datasets. In addition, the maximum sequence length is set to 200 for ML-1M and 50 for the other datasets. Following previous work [38], we use Gurobi software [7] in MIP, limiting the time to 10 minutes with an optimization gap 0.0001 for every user separately. We set the number of candidates to 1,000 and 100 respectively for MIP and MCF, following their settings.

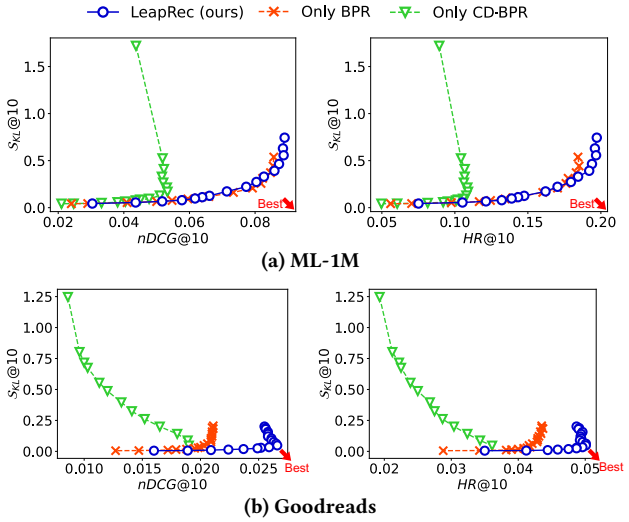
4.2 Performance Comparison (Q1)

4.2.1 Trade-off Comparison. In Figure 5, we compare LeapRec and baselines on four real-world datasets to verify whether LeapRec provides better trade-off between accuracy and calibration than the baselines. We present the performance of the backbone models for both LeapRec (base) and SASRec (base), along with trade-off curves of the methods between accuracy and calibration. The results show that LeapRec consistently surpasses the baselines across all datasets, drawing better trade-off curves. Notably, the observed higher accuracy of LeapRec (base) compared to SASRec



(a) ML-1M (b) Goodreads (c) Grocery (d) Steam

Figure 6: Ablation study of LeapRec. The first row shows $nDCG@10$ vs. $\mathcal{S}_{KL}@10$, and the second row shows $HR@10$ vs. $\mathcal{S}_{KL}@10$. The main components of LeapRec help improve the performance.



(a) ML-1M

(b) Goodreads

Figure 7: Investigation of calibration-disentangled learning-to-rank loss. Using only $\mathcal{L}_{CD-BPR}$ significantly reduces performance compared to using only $\mathcal{L}_{BPR}$ or a combined approach (LeapRec), which demonstrates the importance of integrating both losses to achieve high accuracy and calibration.

(base) shows that calibration-disentangled learning-to-rank offers an enhanced learning mechanism. Additionally, LeapRec maintains high accuracy levels while significantly enhancing calibration, unlike other competing methods. Furthermore, both LeapRec and CaliRec$^+$ achieve the lowest levels of sequential miscalibration by optimizing terms during reranking. This advantage becomes particularly pronounced in datasets with longer user sequences, such as ML-1M and Goodreads, highlighting the importance of addressing sequential miscalibration as user interactions evolve over time.

We further expand our analysis by comparing the trade-off performance on ML-1M and Goodreads datasets using GRU4Rec[5], Caser[6], and BERT4Rec[7] as alternative backbone models. Similar to the results presented in Figure 5, LeapRec outperforms the baselines significantly in terms of trade-off between accuracy and calibration, even when employing different backbone models such as GRU4Rec, Caser, and BERT4Rec.

4.2.2 Complexity Comparison. The model training phase of LeapRec, which incorporates $\mathcal{L}_{BPR}$ and $\mathcal{L}_{CD-BPR}$, maintains a complexity level comparable to traditional methods. Adding $\mathcal{L}_{CD-BPR}$ does not substantially increase the computational burden due to its efficient integration, with the Kullback-Leibler (KL) divergence computation being a $O(d)$ task, where d represents the dimensionality of embedding vectors. The reranking phase is where LeapRec distinctly differs from baselines in terms of time complexity. In Table 2, we compare the computational complexity of reranking algorithms. The table shows that LeapRec and similar greedy approaches like CaliRec and CaliRec$^+$ are notably efficient compared to the more complex algorithms such as MIP and MCF.

4.2.3 Speed Comparison. In Table 3, we compare the speed of LeapRec and baselines by measuring the reranking times across four real-world datasets. The batch size for all tests is standardized at 128 users. For methods like MIP and MCF, 1,000 candidate items per user are considered to ensure a fair comparison. The results in the table confirm LeapRec's competitive speed, comparable to CaliRec and CaliRec$^+$, thereby demonstrating its efficiency in rapidly generating recommendations. Importantly, LeapRec not only matches the speed of CaliRec and CaliRec$^+$ but also significantly surpasses baselines in achieving a superior trade-off between accuracy and calibration as described in Section 4.2.1.

4.3 Ablation Study (Q2)

In Figure 6, we provide an ablation study that compares LeapRec with its variants to evaluate the impact of its core components on performance. The variant *(-)CD* removes calibration-disentangled

[5]https://github.com/jeon185/LeapRec/blob/main/experiments/GRU4Rec.pdf

[6]https://github.com/jeon185/LeapRec/blob/main/experiments/Caser.pdf

[7]https://github.com/jeon185/LeapRec/blob/main/experiments/BERT4Rec.pdf

learning, by adopting a naive SASRec instead of our proposed training approach as the backbone. The variants *(-)PR-uniform* and *(-)PR-reverse* remove Property 1 (i.e., relevance priority property) from the objective function in the reranking phase. Specifically, *(-)PR-uniform* applies a uniform balancing coefficient λ, diverging from the adaptive $\lambda^{1/k}$ used in Equation (11). In contrast, *(-)PR-reverse* inverses this adaptation by employing λ^k, prioritizing calibration in higher-ranked recommendations. Hence, *(-)PR-uniform* considers the relevance and calibration equally in all k'th recommendations, whereas *(-)PR-reverse* prioritizes the calibration in the higher-ranked recommendations. The results show the superiority of LeapRec over its variants, verifying the effectiveness of its core components. Notably, LeapRec shows better performance than *(-)CD* in most cases, indicating that disentangling calibration during the training of personalized rankings is essential for achieving both high accuracy and calibration. In addition, LeapRec consistently outperforms *(-)PR-uniform* and *(-)PR-reverse* in achieving better balances between accuracy and calibration across most cases. Especially, *(-)PR-reverse*, which deliberately inverts the relevance priority property, exhibits a more pronounced decline in performance compared to *(-)PR-uniform*, which merely omits the property; these observations validate the significance of entailing the relevance priority property in balancing accuracy and calibration.

In Figure 7, we further explore the impact of exclusively using $\mathcal{L}_{CD-BPR}$, without using $\mathcal{L}_{BPR}$, on ML-1M and Goodreads datasets. This analysis is presented separately due to the significantly different performance scales observed. The variant *Only BPR*, akin to *(-)CD*, employs only $\mathcal{L}_{BPR}$ to train the backbone model. In contrast, the variant *Only CD-BPR* is trained solely with $\mathcal{L}_{CD-BPR}$. The results show that *Only CD-BPR* substantially underperforms compared to LeapRec and *Only BPR*. This outcome supports the discussion in Section 3.1 about the need to combine $\mathcal{L}_{BPR}$ and $\mathcal{L}_{CD-BPR}$ to achieve high performance both on accuracy and calibration.

4.4 Effect of the balancing hyperparameter (Q3)

The balancing hyperparameter λ, as defined in Equation (11), is a key factor in controlling the balance between accuracy and calibration. Increasing λ leads to more calibrated recommendations at the cost of accuracy. In Figure 8, we examine the impact of varying λ on users' overall recommendation quality on Goodreads dataset. We use kernel density estimation (KDE) to evaluate the distribution of sequential miscalibration $\mathcal{S}_{KL}@10$ across users under different λ. In the figure, we also denote the average nDCG@10 to show the change in accuracy. The results of low λ (e.g., 0.1) or missing calibration (LeapRec (base)) show a wide range of distribution for $\mathcal{S}_{KL}@10$, meaning a substantial variance in calibration across users. However, as λ increases, recommendations become more uniformly calibrated across users. The accuracy increases until λ reaches 0.7.

4.5 Case Study (Q4)

In Figure 9, we analyze a case to observe how LeapRec balances relevance and calibration when recommending a list of items to a user on Grocery dataset. We vary the calibration level for a random user and observe how the user experiences the recommendations. Figure 9 (a) represents the user's sequential interactions and the ground-truth item that the user will interact with. Figures 9 (b-d) show the recommendation results of LeapRec for λ (defined in Equation (11)) values 0.0, 0.3, and 0.9, respectively. Figure 9 (b)

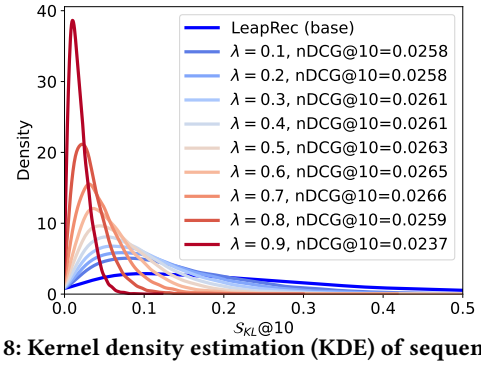


Figure 8: Kernel density estimation (KDE) of sequential miscalibration $\mathcal{S}_{KL}@10$ on Goodreads dataset, while varying the balancing hyperparameter λ; range of the x-axis is set to $[0, 0.5]$. Averaged nDCG@10 for each setting is also written in the legend. The wide range of KDE indicates the recommendation quality in terms of calibration varies across users. LeapRec successfully provides higher calibrated recommendations to more users as λ increases.

shows that LeapRec accurately recommends the ground-truth item at rank 1 without calibration (i.e., $\lambda = 0.0$). However, the overall items in the list skew towards the user's major interest *Beverage*, narrowing the user experience for the recommendation. In Figures 9 (c-d), we observe that LeapRec effectively considers the user's other interest *Candy & Chocolate* as well as the major interest *Beverage* by enhancing calibration. Notably, LeapRec effectively retains *English breakfast tea*, the most relevant item, at rank 1 while improving calibration. We further analyze another case where a user's category preference shifts towards a category previously not favored[8].

5 RELATED WORKS

Calibrated recommendation. Steck [39] first introduced calibrated recommendation to address the problem of traditional recommendation, where less dominant interests of users are often neglected. The goal is to ensure that recommendation lists accurately reflect the proportion of categories users have shown interest in. Steck [39] proposed CaliRec which is a post-processing approach that greedily adjusts the recommended items to better match the user's historical category distribution. Subsequent studies continued to adopt the post-processing strategy. Seymen et al. [38] introduced a non-greedy approach by formulating calibration as a constrained optimization problem and solving it with a mixed integer programming (MIP) algorithm. Abdollahpouri et al. [1] defined the calibrated recommendation as the maximum flow optimization problem and proposed a minimum cost flow (MCF) based algorithm. On the other hand, Chen et al. [3] proposed DACSR, an end-to-end method, to simultaneously optimize accuracy and calibration in a single training phase.

Different from earlier advancements, our proposed method directly integrates calibration scores within the training phase, using a novel loss designed to enhance both the training and post-processing phases. Additionally, we introduce a prioritized mechanism to effectively balance accuracy and calibration, addressing the dual criteria challenge more dynamically than previous methods.

[8] https://github.com/jeon185/LeapRec/blob/main/experiments/CaseStudy.pdf

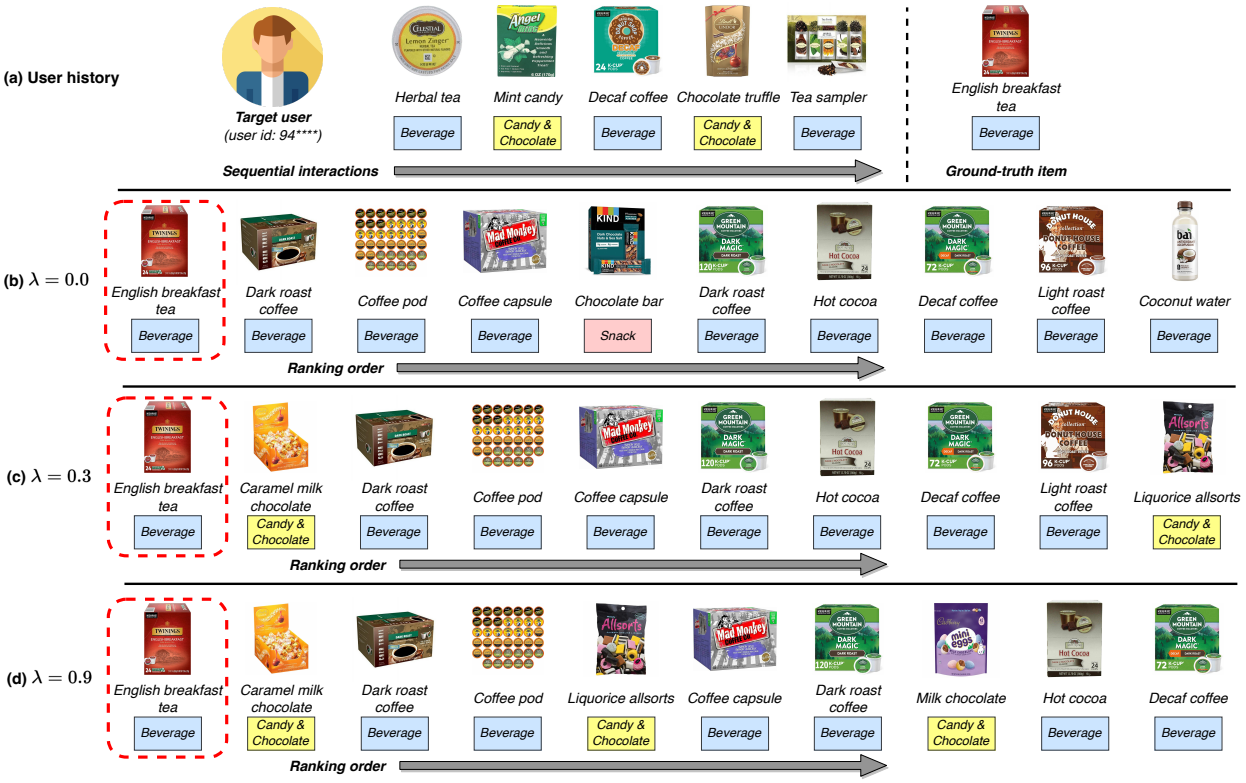


Figure 9: (a) A real user's sequential interactions and ground-truth next item in Grocery dataset. (b-d) The top-10 recommended items for the user by LeapRec with different λ: 0.0, 0.3, and 0.9. We mark the ground-truth item with a red box in each recommendation result. Colored boxes denote the categories. The values of $\mathcal{S}_{KL}$@10 are 1.5920, 0.1001, and 0.0212 respectively for $\lambda = 0.0$, $\lambda = 0.3$, and $\lambda = 0.9$. This case shows that LeapRec enhances the calibration of the overall recommendation list while keeping relevant items at the top rank, as λ increases.

Sequential recommendation. Sequential recommender systems predict users' future interactions by considering the temporal dynamics of their sequential interactions. Such systems have shown their effectiveness by capturing the users' long-term and short-term interests [13, 36]. As early work, FPMC [36] integrated first-order Markov chains with matrix factorization techniques [25, 37] to simultaneously address users' sequential activities and overall preferences. Subsequent research has evolved to include higher-order Markov chains [9, 10], capturing more complex sequential dependencies by considering multiple preceding interactions. Within the last decade, deep learning models such as Recurrent Neural Networks (RNN) [6, 14], Convolutional Neural Networks (CNN) [26], and Transformers [42] have been adopted in sequential recommendation, marking significant progress through their ability to model non-linear relationships in user behaviors. Models such as GRU4Rec [13] and GRU4Rec$^+$ [12] demonstrated the effectiveness of GRU [6] in session-based recommendations. Tang and Wang [41] effectively utilized CNN for extracting sequential patterns across both temporal and feature dimensions. More recently, Transformer-based models such as SASRec [20], BERT4Rec [40], and SmartSense [19] leveraged unidirectional and bidirectional Transformers to capture complex correlations within a sequence. Other techniques include memory networks [4, 16], translation learning [9], hierarchical attention learning [49], graph neural network learning [2, 30, 47], and contrastive learning [5, 48, 51].

We focus on integrating calibration within sequential recommender systems. Our method is model-agnostic, allowing it to be applied across various sequential recommender systems.

6 CONCLUSION

In this work, we propose LeapRec, a novel method that effectively balances accuracy and calibration in sequential recommendation. LeapRec first trains a backbone model using the proposed calibration-disentangled learning-to-rank loss to learn personalized rankings when calibration is considered. Subsequently, LeapRec applies the proposed relevance-prioritized reranking algorithm to the backbone's results, encouraging highly relevant items are placed at the top while accounting for calibration throughout the recommendations. LeapRec achieves superior performance over existing calibrated recommendation methods in extensive experiments. Our ablation study further confirms the necessity of the core components of LeapRec. We also demonstrate through a case study how LeapRec tackles relevance and calibration to achieve high performance on both.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00244831).

REFERENCES

[1] Himan Abdollahpouri, Zahra Nazari, Alex Gain, Clay Gibson, Maria Dimakopoulou, Jesse Anderton, Benjamin A. Carterette, Mounia Lalmas, and Tony Jebara. 2023. Calibrated Recommendations as a Minimum-Cost Flow Problem. In *WSDM*.

[2] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential Recommendation with Graph Neural Networks. In *SIGIR*. ACM.

[3] Jiayi Chen, Wen Wu, Liye Shi, Yu Ji, Wenxin Hu, Xi Chen, Wei Zheng, and Liang He. 2022. DACSR: Decoupled-Aggregated End-to-End Calibrated Sequential Recommendation. *Applied Sciences* (2022).

[4] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks. In *WSDM*. ACM.

[5] Yongjun Chen, Zhiwei Liu, Jia Li, Julian J. McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *WWW*. ACM.

[6] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*. ACL.

[7] LLC Gurobi Optimization. 2021. Gurobi Optimizer Reference Manual. https://www.gurobi.com.

[8] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* (2016).

[9] Ruining He, Wang-Cheng Kang, and Julian J. McAuley. 2017. Translation-based Recommendation. In *RecSys*. ACM.

[10] Ruining He and Julian J. McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *ICDM*. IEEE Computer Society.

[11] Ruining He and Julian J. McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*. ACM.

[12] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *CIKM*. ACM.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR*.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* (1997).

[15] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*. IEEE Computer Society.

[16] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *SIGIR*. ACM.

[17] Hyunsik Jeon, Jun-Gi Jang, Taehun Kim, and U Kang. 2022. Accurate Bundle Matching and Generation via Multitask Learning with Partially Shared Parameters. *CoRR* (2022).

[18] Hyunsik Jeon, Jongjin Kim, Jaeri Lee, Jong-eun Lee, and U Kang. 2023. Aggregately Diversified Bundle Recommendation via Popularity Debiasing and Configuration-Aware Reranking. In *PAKDD*. Springer.

[19] Hyunsik Jeon, Jongjin Kim, Hoyoung Yoon, Jaeri Lee, and U Kang. 2022. Accurate Action Recommendation for Smart Home via Two-Level Encoders and Commonsense Knowledge. In *CIKM*. ACM.

[20] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. IEEE Computer Society.

[21] Jongjin Kim, Hyunsik Jeon, Jaeri Lee, and U Kang. 2023. Diversely Regularized Matrix Factorization for Accurate and Aggregately Diversified Recommendation. In *PAKDD*. Springer.

[22] Anton Klenitskiy and Alexey Vasilev. 2023. Turning Dross Into Gold Loss: is BERT4Rec really better than SASRec?. In *RecSys*. ACM.

[23] Bonhun Koo, Hyunsik Jeon, and U Kang. 2020. Accurate News Recommendation Coalescing Personal and Global Temporal Preferences. In *PAKDD*. Springer.

[24] Bonhun Koo, Hyunsik Jeon, and U Kang. 2021. PGT: news recommendation coalescing personal and global temporal preferences. *Knowl. Inf. Syst.* (2021).

[25] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* (2009).

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*.

[27] Wonbin Kweon, SeongKu Kang, and Hwanjo Yu. 2022. Obtaining Calibrated Probabilities with Personalized Ranking Models. In *AAAI*. AAAI Press.

[28] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *CIKM*. ACM.

[29] Jiacheng Li, Yujie Wang, and Julian J. McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *WSDM*. ACM.

[30] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory Augmented Graph Neural Networks for Sequential Recommendation. In *AAAI*. AAAI Press.

[31] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A Simple and Strong Baseline for Collaborative Filtering. In *CIKM*. ACM.

[32] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*. ACM.

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*.

[34] Apurva Pathak, Kshitiz Gupta, and Julian J. McAuley. 2017. Generating and Personalizing Bundle Recommendations on *Steam*. In *SIGIR*. ACM.

[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. AUAI Press.

[36] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *WWW*. ACM.

[37] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NIPS*.

[38] Sinan Seymen, Himan Abdollahpouri, and Edward C. Malthouse. 2021. A Constrained Optimization Approach for Calibrated Recommendations. In *RecSys*.

[39] Harald Steck. 2018. Calibrated recommendations. In *RecSys*.

[40] 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 *CIKM*. ACM.

[41] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*. ACM.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.

[43] Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *RecSys*. ACM.

[44] Lequn Wang and Thorsten Joachims. 2023. Uncertainty Quantification for Fairness in Two-Stage Recommender Systems. In *WSDM*. ACM.

[45] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet A. Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. In *IJCAI*.

[46] Penghui Wei, Weimin Zhang, Ruijie Hou, Jinquan Liu, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. Posterior Probability Matters: Doubly-Adaptive Calibration for Neural Predictions in Online Advertising. In *SIGIR*. ACM.

[47] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. In *AAAI*. AAAI Press.

[48] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. In *ICDE*. IEEE.

[49] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *IJCAI*. AAAI Press.

[50] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *WSDM*. ACM.

[51] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. CauseRec: Counterfactual User Sequence Synthesis for Sequential Recommendation. In *SIGIR*. ACM.