Personalized Diversification for Neural Re-ranking in Recommendation

Weiwen Liu*
Huawei Noah's Ark Lab
liuweiwen8@huawei.com

Yunjia Xi*
Shanghai Jiao Tong University
xiyunjia@sjtu.edu.cn

Jiarui Qin
Shanghai Jiao Tong University
qjr1996@sjtu.edu.cn

Xinyi Dai
Shanghai Jiao Tong University
daixinyi@sjtu.edu.cn

Ruiming Tang†
Huawei Noah's Ark Lab
tangruiming@huawei.com

Shuai Li
Shanghai Jiao Tong University
shuaili8@sjtu.edu.cn

Weinan Zhang
Shanghai Jiao Tong University
wnzhang@sjtu.edu.cn

Rui Zhang†
ruizhang.info
rayteam@yeah.net

***Abstract*—Re-ranking, as the final stage of the multi-stage recommender systems (MRS), aims at modeling the listwise context and the cross-item interactions between the candidate items. The objective is usually the overall utility (*e.g.*, total clicks or revenue) of the re-ranked list, which is determined not only by the relevance, but also by the diversity of the list. However, existing methods equally promote diversity for all users and often compromise the relevance ranking. In reality, users have different diversity preferences and we should diversify the list tailored to individual users' interests and needs. Users' behavior history contains rich information which may be used for inferring their diversity preferences, but has rarely been explored in existing work. In this work, we propose a novel neural re-ranking with personalized diversification method (dubbed RAPID) to address the above challenge. RAPID explicitly models each user's preference distribution over different topics by exploiting the intra- and inter-topic interactions from the user's behavior history. The personalized diversity gain brought by each candidate item is then measured by the item's marginal diversity and the learned personalized preference. The relevance and the personalized diversity are jointly optimized in an end-to-end manner to automatically manage the relevance-diversity tradeoff. Experimental results on two public datasets and a proprietary dataset show that RAPID outperforms the state-of-the-art with the highest utility and the best relevance-diversity tradeoff. We further prove that RAPID has a regret bound of $\tilde{O}(\sqrt{n})$ on utility, which provides theoretical guarantee that its performance is near-optimal.**

I. Introduction

Multi-stage Recommender Systems (MRS) are widely adopted by many large online platforms [1]–[4], where the recommendation process is split into multiple stages to meet the requirement of short response time. Typical stages in MRS includes recall (*a.k.a*, matching), ranking, and re-ranking [5]. Each stage narrows down the relevant items with a slower but more accurate model [6]. The re-ranking stage, as the final stage in MRS, has a direct impact on user experience and satisfaction, and thus plays a critical role in recommendation. The re-ranking stage focuses on modeling listwise context/cross-item interactions between candidates, where whether an item is relevant to a user is not only determined by the item itself, but also depends on other items placed in the same list [2], [5], [7]. The overall listwise relevance is the combined effect of all the items in the list.

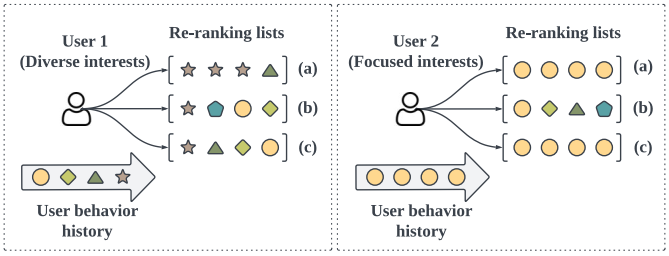

Fig. 1: A motivating example of different re-ranking strategies: (a) purely relevance-oriented re-ranking; (b) traditional diversity-aware re-ranking; (c) re-ranking with personalized diversification. Different shapes denote different topics.

Recent re-ranking evolves to the deep neural architectures and its performance is greatly benefited from the automatic learning of the listwise context [2], [7]–[9]. However, most neural re-ranking models are purely relevance-oriented[1] [1], [2], [4], [7], and often lead to similar or near-duplicate re-ranked results [9], [10]. The lack of exploration of diverse items can hurt users' satisfaction and cause the filter bubble effect [9]–[12]. To diversify the recommendation results, several neural diversity-aware models have been proposed [9], [11], [13], [14]. While re-ranked lists are more diverse, users' **personal preferences for diversity** are overlooked.

In fact, diversification for re-ranking should be tailored and personalized. Users' preferences over different topics (*e.g.*, categories or genres of items) vary greatly [15], [16]. For example, some users may have a narrow taste and highly prefer a particular type of music, say a classical music lover, while others may be open-minded and enjoy diverse styles of music. Therefore, assuming users' diversity preferences are the same and equally increasing diversity for all users may hurt users' satisfaction, especially for users with focused

*Equal contribution.
†Corresponding authors.

[1]In this work, *relevance* represents how likely an item is found attracted, and *utility* is a listwise metric and measures the total click or income of a list, depending on relevance, list diversity, and other context factors.

interests. Instead, diversification should suit individual user's diversity preference. Figure 1 illustrates how re-ranking can be benefited from personalized diversification. User 1 has broad interests and favored multiple topics in the past (marked by different shapes in her behavior history), while user 2 has focused interests and was attracted by only one topic. A purely relevance-oriented re-ranking model (a) tends to recommend a list of items from similar topics, ignoring user 1's diverse tastes. For traditional diversity-aware models (b), the re-ranked lists are highly diversified but present users with uninterested topics, which sacrifices the total utility. In contrast, re-ranking with personalized diversification (c) diversifies the lists as per the user's interests over different topics, showing the potential to jointly optimize relevance and diversity. Specifically, on the one hand, if most items in the list are similar as in (a), users probably only click the one most relevant item and yield relatively limited overall utility. A diverse list that fits users' complete interests as in (c) can fulfill users' multiple information needs from different aspects, and thus encourages multiple clicks. On the other hand, providing diverse recommendation to users with diverse interests improves the overall diversity.

Moreover, we find that users' behavior history carries rich information for inferring their "true" preference for diversity, but has rarely been studied in existing work. A user's behavior history contains sequences of items that were positively interacted with by the user (*e.g.*, clicked, browsed, purchased), and has been widely used for user modeling in recommendation [17], [18]. Existing re-ranking models [3], [19], however, generally transform users' history into a fixed-length vector as extra user features, unaware of the hidden diversity preference for individual users. The behavior history over different topics records users' diverse needs, and suggests their personal intents in different aspects. But how to effectively uncover the correlation between the sequences of users' behavior history and the current ranking list for re-ranking remains a challenging problem. For one thing, users' behaviors for different topics are mixed up in one long sequence, and is difficult to directly extract useful information from it. For another, users' preferences for different topics are correlated, and thus it is important to identify the inter-dependencies between topics.

To address the above problems, we propose a new neural re-ranking with personalized diversification method, RAPID. The key idea lies in adaptively learning the user's personalized diversity interests from her behavior history sequences, and providing diverse re-ranked lists catered to each user's tastes and preferences. In particular, we design a *personalized diversity estimator* to capture the intra-topic and inter-topic interactions from the user's behavior history, and learn the diversity gain for each item conforming to the user's personal interests. *The listwise relevance estimator* captures the cross-item interactions between candidate items. The tradeoff between the personalized diversity and the relevance is automatically optimized to generate the final re-ranking list. The main contributions of this paper are summarized as follows:

- We propose to model the personalized diversity preference via users' behavior history in RAPID, and provide diverse re-ranked lists catered to each user's preferences. Both intra- and inter-topic interactions are incorporated to learn individual users' preference distribution from their behavior history. To the best of our knowledge, this is the first work that automatically learns personalized diversity preferences for the re-ranking stage.
- We jointly encode the item relevance and personalized diversity for re-ranking, which enables automatic optimization of the tradeoff between them. Hence RAPID is aware of the diversity need for different recommendation scenarios without manual intervention.
- We theoretically analyze the efficacy and efficiency of RAPID. For efficacy, we show that the performance of RAPID is guaranteed to produce a near-optimal recommendation, with a $\tilde{O}(\sqrt{n})$ regret bound on utility. For efficiency, we show that the computation cost of RAPID suffices the response time requirement of large industrial recommender systems.

Extensive experiments on both semi-synthetic datasets and a real-world industrial dataset show that RAPID outperforms the state-of-the-art re-ranking methods and yields the highest utility and the best relevance-diversity tradeoff.

II. Related Work

A. Relevance-oriented Re-ranking

Several existing studies have shown the effectiveness of re-ranking for relevance by automatically modeling listwise context/cross-item interactions with deep neural networks [1]–[3], [5], [7], [19]–[22]. For example, DLCM [7] uses recurrent neural networks to sequentially encode the top-ranked items with their feature vectors. Inspired by the transformer architecture used in machine translation [23], PRM [2] applies the transformer to model the cross-item interactions. Pang *et al*. [20] study the permutation-invariant property of the self-attention blocks in re-ranking and propose SetRank. SRGA [19] advances the attention structure by considering the unidirectional browsing behavior and the local interaction between neighboring items. Yet all of them focus on relevance and overlook users' demands for diversity.

B. Diversity-aware Re-ranking

To obtain diversified recommendation results, a series of re-ranking models have been proposed [9]–[11], [13], [14], [24]–[29]. An early example is the Maximum Marginal Relevance (MMR) algorithm [24], [25], which diversifies the list by greedily adding items with maximal marginal relevance. The Determinantal Point Process (DPP) [10], [28] is recently introduced to recommender systems by defining a probability distribution over subsets of items. SSD [29] improves the diversity by maximizing the volume spanned by the constructed trajectory tensor with sliding windows. As for neural methods, NTN-DIV [14] proposes a neural tensor network to learn the dissimilarity between any pairs of items. DSSA [13] estimates the diversity with a recurrent network, and uses a weighted combination of the relevance and diversity to produce the next

TABLE I: Notations and descriptions.

Notations	Descriptions
$u, \mathcal{U}$	user u from the user set $\mathcal{U}$
$v, \mathcal{V}$	item v from the item set $\mathcal{V}$
x_u/x_v	features for user/item
q_u/q_v	dimension of user/item features
R	initial ranking list, $R(i)$ is the i-th item in the list
L	length of the initial list
S	re-ranked list
K	length of the re-ranked list
m	number of topics
$c(G)$	topic coverage of a set/list G, $c_j(G)$ is its j-th element
$\mathcal{T}$	user's behavior sequence
D	maximum length of the user's behavior sequences
ϕ_R	re-ranking scores for list R, $\phi_R \in \mathbb{R}^L$
$\tilde{\phi}_R$	underlying ground truth attraction probability for list R
$\tilde{\phi}$	attraction probability estimated by DCM for click generation
$\tilde{\alpha}$	relevance estimated by DCM for click generation
$\tilde{\epsilon}$	termination probability estimated by DCM for click generation

item. DESA [9] captures item diversity and relevance with a self-attention structure, and Le *et al*. [11] propose to maximize a differentiable diversity-aware loss. However, existing studies diversify the list equally for all users, and have not addressed users' personal preferences towards the diversity of items.

C. Personalized Diversification

Personalized diversification, taking advantage of both personalization and diversification, provides tailored diversity for individual users. Early personalized diversification is heuristic and computed according to specific statistics. Shi *et al*. [30] use the variance of the latent user factors to indicate a user's demand for diversity. Wu *et al*. [31] find that the big-five personality model (e.g., conscientiousness) is correlated with users' diversity needs and can be used to improve the diversity. But it assumes the personality values of each user in terms of the five personality dimensions (extraversion, agreeableness, openness, conscientiousness, and neuroticism) are known, which is hard to collect for most of today's recommender systems. Conducting surveys to obtain the personality values requires extra responses from users beforehand and can cause inconvenience to users. Noia *et al*. [32] compute the user's propensity toward diversity by the entropy over different categories and the profile length. These methods are rule-based and non-learnable, which may be biased to the manually designed rules and lack generalization ability. PD-GAN [12], proposes to learn a personalized DPP kernel matrix with adversarial learning. However, PD-GAN is designed for the ranking stage in recommendation, and ignores the complicated cross-item interaction for relevance. Moreover, PD-GAN is a two-stage model that first estimates relevance scores and then diversifies the results. In this decoupled learning process of diversity and relevance, the diversification signal cannot be reflected in the relevance models, which results in suboptimal recommendation. In this work, we jointly learn the relevance and the personalized diversity scores, while modeling the listwise cross-item interactions.

III. Proposed Method

In this section, we first give the problem formulation and an overview of RAPID in Section III-A, followed by the detailed description in Sections III-B to III-E.

A. Problem Formulation

The goal of re-ranking with personalized diversification is to optimize the overall utility and generate re-ranked lists that are both relevant and diverse according to users' personal preferences. Suppose we have a set of users $\mathcal{U}$ and a set of items $\mathcal{V}$, and items belong to a set of m topics $\mathcal{J} = \{1, \dots, m\}$. Each item $v \in \mathcal{V}$ is associated with a *topic coverage* feature $\tau_v \in \mathbb{R}^m$, with the j-th element $\tau_v^j \in [0, 1]$ denotes the probability[2] of item v covering topic $j \in \mathcal{J}$. For a specific user $u \in \mathcal{U}$, given an initial ranking list R_u of L items, the re-ranking model aims to learn a multivariate scoring function $\mathcal{F}$ defined on R_u and output the re-ranking scores ϕ_{R_u} that consider both relevance and diversity. The top-K items with the highest re-ranking scores are selected as the re-ranked list S_u and finally displayed to the user, usually $K \leq L$.

Formally, the input domain for the scoring function $\mathcal{F}$ is $\Omega = \{R_u, \tau_{R_u}, \mathcal{T}_u | u \in \mathcal{U}\}$, where τ_{R_u} is the topic coverage information of the initial list R_u (stacking all τ_v, $v \in R_u$) and $\mathcal{T}_u$ is the behavior sequence for user u, then the scoring function $\mathcal{F} : \Omega \to \mathbb{R}^L$ is defined as

$$\phi_{R_u} = \mathcal{F}(R_u, \tau_{R_u}, \mathcal{T}_u) = g\left(f_r(R_u, \mathcal{T}_u), f_d(\tau_{R_u}, \mathcal{T}_u)\right) , \quad (1)$$

where ϕ_{R_u} represents the estimated utility, f_r and f_d are the relevance function and the personalized diversity function, respectively, and g is an aggregation function that combines the estimated relevance and personalized diversity. In particular, the diversity term f_d is measured by the topic information τ_{R_u} and the personalized preferences that adaptively learned from historical behaviors $\mathcal{T}_u$, varying for different users. This is in comparison to indiscriminately computing the diversity only using τ_{R_u} as in [9], [13]. Table I shows the main notations and their descriptions used in this paper. For simplicity, we omit the subscript u from subsequent notations (*e.g.*, ϕ_{R_u} to ϕ_R), if not otherwise specified.

Overview. Figure 2 shows the overall framework of RAPID, which is consist of a *listwise relevance estimator*, a *personalized diversity estimator*, and a *re-ranker*. The listwise relevance estimator takes as input the user and item features of the initial ranking list and estimates the listwise context. The personalized diversity estimator extracts users' individual preference distribution over different topics from their behavior history, and outputs the personalized diversity gain of the candidate items by combining the topic coverage information and the preference distribution. Finally, the estimated listwise relevance and the personalized diversity of candidate items are aggregated together by our proposed deterministic or probabilistic approach to predict the re-ranking scores in the re-ranker module. Therefore, the relevance and the personalized

[2]Here, we present the item coverage τ_v^j in probability to be more general. Usually, $\tau_v^j \in \{0, 1\}$ is binary and indicates if item v belongs to topic j.

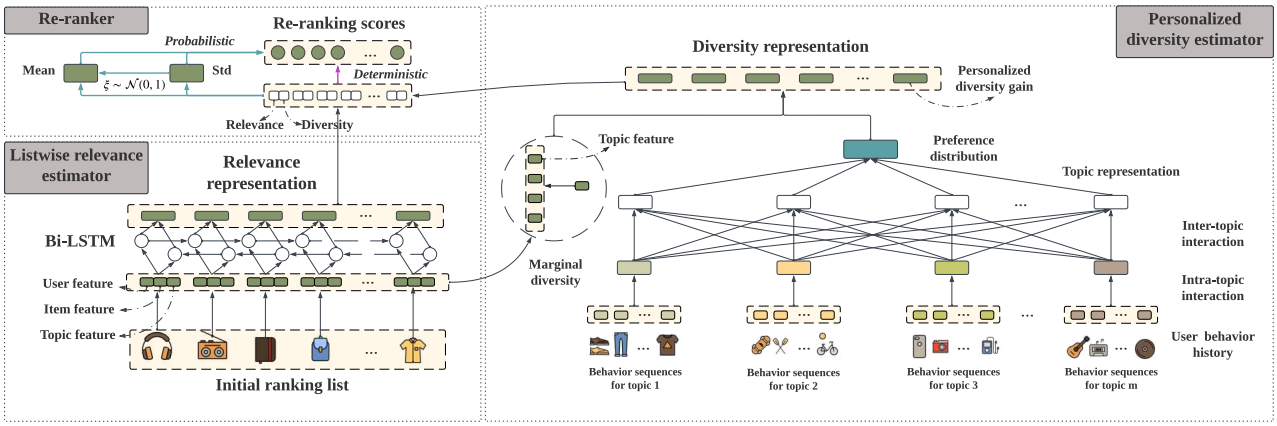

Fig. 2: Overall framework of RAPID, which jointly estimates the listwise context and the personalized diversity. The purple and blue arrows in the re-ranker module represent the deterministic and the probabilistic output approach, respectively.

diversity are jointly trained in an end-to-end manner. Details of each component are elaborated on in the following sections.

B. Listwise Relevance

The relevance of an item is not independent, rather it is affected by other items placed in the same list. For example, placing complementary products in one list, like a phone and a phone case, usually receives higher total attention [33]. Re-ranking is designed to model such cross-item interactions/listwise context [5]. In particular, the user's interests in the current item are influenced by items ranked both before and after it. Therefore, we adopt a bidirectional LSTM (Bi-LSTM) [34] to bidirectionally exploit such listwise context and the sequential dependencies. Bi-LSTM has exhibited great effectiveness in modeling the two-way long short-term dependencies of a list [35].

Let $R(i)$ denote the i-th item in the initial ranking list R, $i = 1, \ldots, L$. Each item $R(i)$ is represented by a concatenation of the user feature x_u, the item feature $x_{R(i)}$, and the topic coverage feature $\tau_{R(i)}$, *i.e.*, $e_{R(i)} = [x_u, x_{R(i)}, \tau_{R(i)}] \in \mathbb{R}^{q_u+q_v+m}$. Then the forward output state $\overrightarrow{h}_{R(i)} = \text{LSTM}(e_{R(i)}, h_{R(i-1)})$ of the i-th item can be acquired by an LSTM cell [36]. Similarly, we can obtain the backward output state $\overleftarrow{h}_{R(i)} = \text{LSTM}(e_{R(i)}, h_{R(i+1)})$. Then the final relevance representation $h_{R(i)}$ for item i is a concatenation of the forward and the backward output states $h_{R(i)} = [\overrightarrow{h}_{R(i)}, \overleftarrow{h}_{R(i)}] \in \mathbb{R}^{2q_h}$, where q_h is the hidden size. As such, RAPID is capable of modeling the relationship between each candidate item and each topic in the initial ranking list for a user. Note that the Bi-LSTM could be replaced by other deep architectures like transformer [2] to model the cross-item interactions. Here, we adopt Bi-LSTM due to its simplicity and relatively superior performance compared to transformer in our experiments (Section IV-E2).

C. Personalized Diversity

Personalized diversification tailors how to diversify the re-ranking list for each user, as different users may have different preferences for diversity. Some users may prefer a diverse recommendation result; some may be inclined to only diversify for specific topics; while others may be only willing to receive focused recommendation. Providing diverse re-ranked lists catered to each user's tastes fulfills the user's multi-faceted interests, thereby could jointly improve the relevance and the diversity. Moreover, the user's history contains rich personalized behavioral characteristics and intents, which is critically important for modeling user interests and diversity preferences. Existing diversity-aware work, however, generally diversifies the recommendation lists uniformly and fails to extract the topic-level fine-grained preferences from the user's history [9], [13]. In RAPID, we first learn the *preference distribution* over different topics from the user's history behavior sequences, and then design a user-specific diversity representation of each candidate item with personalized diversity gain to facilitate diverse recommendation.

Since items from different topics are mixed up in users' behavior history, we split the whole long sequence of history and construct separate behavior sequences for every topic $\mathcal{T}_1, \ldots, \mathcal{T}_m$ to better model the user's preferences, as shown in Figure 2. Each topical behavior sequence $\mathcal{T}_j$ containing items favored by the current user belonging to topic $j \in \mathcal{J}$, sorted by time. Whether an item belongs to a topic can be sampled according to its given topic coverage. Let D be the maximum length of the behavior sequences. We incorporate LSTM to explicitly encode time dependencies of user behaviors within each topic to model the intra-topic interactions. Let $\mathcal{T}_j(i)$ be the i-th item in sequence $\mathcal{T}_j$. Then the output states are derived by $z_{j,i} = \text{LSTM}([x_u, x_{\mathcal{T}_j(i)}], z_{j,i-1})$, which take as input the concatenated user and item features $[x_u, x_{\mathcal{T}_j(i)}]$ and the previous states.

The final output state of LSTM can be viewed as an encoding of all the information contained in the sequence, and has the greatest impact on future predictions. Thus we take the final output states as the representation vector for each topic j, *i.e.*, $t_j = z_{j,D}$. The topic representation vector t_j summarizes the user's interests in topic j, which is personalized and beneficial for the final re-ranking.

After encoding users' behavioral patterns separately for each topic, we aggregate topic-specific signals to obtain the *preference distribution* over topics by self-attention mechanism [23]. The self-attention mechanism is used for capturing

the relationship and correlations between different topics—the inter-topic interactions. We stack all topic representation vectors $t_j, j = 1, \ldots, m$ and obtain a matrix $V \in \mathbb{R}^{m \times q_h}$ with q_h as the hidden size. The self-attention is defined by

$$A = \text{Attention}(V) = \text{softmax}\left(\frac{VV^\top}{\sqrt{q_h}}\right)V\,, \tag{2}$$

where $A \in \mathbb{R}^{m \times q_h}$ is the attended matrix, and $\sqrt{q_h}$ is used to stabilize gradients during training. We denote the j-th row of A as a_j, and further feed concatenated $[a_1, \ldots, a_m]$ to a multi-layer perceptron (MLP) by non-linearly mapping the attended vectors into an m-dimensional space to generate the personalized preference distribution $\hat{\theta} \in \mathbb{R}^m$ over topics,

$$\hat{\theta} = \text{MLP}_\theta[a_1, \ldots, a_m]\,, \tag{3}$$

where $\hat{\theta}$ describes how likely a user is interested in each topic.

Moreover, given the initial ranking list R, the diversity of an item is not independent, but relies on the dissimilarity or novelty between the current item and other items in R. In this paper, we use the probabilistic coverage function $c(\cdot)$ as the diversity function, which is a commonly adopted submodular function for recommendation diversity [26], [37]–[39],

$$c_j(R) = 1 - \prod\nolimits_{v \in R}(1 - \tau_v^j)\,, \tag{4}$$

where τ_v^j is the item's topic coverage, and $c_j(R)$ describes the probability of at least one item in R covers topic j. Notice that the probabilistic coverage function can be replaced by other submodular diversity functions according to the objective of the recommendation scenario.

Furthermore, we define the *marginal diversity* of an item $R(i)$ as the difference in diversity with and without $R(i)$ appearing in the list,

$$d_R(R(i)) = c(R) - c(R/\{R(i)\})\,, \tag{5}$$

where $d_R(R(i)) \in [0,1]^m$, $R/\{R(i)\}$ refers to the remained list with $R(i)$ removed. The quantity in Eq.(5) indicates the dissimilarity between $R(i)$ and all the other candidate items in R. Clearly, if an item is similar to other candidate items or its characteristics have already been covered by the combination of other items in the list, adding it to the list will bring about little improvement in list diversity.

Building upon the previous discussion, we can attain the diversity representation vector $\Delta_R(R(i))$ of each item $R(i), i = 1, \ldots, L$ from the initial ranking list R,

$$\Delta_R(R(i)) = \hat{\theta} \odot d_R(R(i))\,, \tag{6}$$

where $\odot$ is the element-wise product. Hence, the j-th element of $\Delta_R(R(i))$ represents the *personalized diversity gain* of item $R(i)$ in topic j, weighted by the user preference for topic j. Intuitively, if item $R(i)$ is more diverse over other items placed in the list and belongs to the user's interested topic, then that item is likely to attract the user and gain more diversity. For example, suppose we have three topics, the current list has already covered topic 1, and the user is also interested in topic 2 but not topic 3, then an item covering topic 2 will bring larger diversity gain to the list compared to the item in topic 3.

D. Re-ranking

To provide the final re-ranking scores, rather than explicitly balance relevance and diversity with a hyper-parameter to be tuned, we let the model automatically fuse the relevance and diversity signals with an MLP. This is in contrast to existing learning methods [11], [29] that require extra manual work of parameter tuning or optimal list construction. To this end, our proposed RAPID is more flexible and can easily adapt to different recommendation scenarios without the heavy burden of manually managing the tradeoff.

We stack the relevance representation vectors $h_{R(i)}$, $i = 1, \ldots, L$ into a matrix $H_R \in \mathbb{R}^{L \times 2q_h}$, and the diversity representation vectors $\Delta_R(R(i))$, $i = 1, \ldots, L$ into a matrix $\Delta_R \in \mathbb{R}^{L \times m}$. Here, we present two output approaches, a deterministic and a probabilistic approach, to produce the final re-ranking scores.

1) Deterministic Approach: We concatenate the relevance matrix and the diversity matrix, and then use an MLP for fusing the relevance and diversity to predict the scores,

$$\phi_R = \text{MLP}_\phi[H_R, \Delta_R]\,, \tag{7}$$

where $\phi_R \in \mathbb{R}^L$, and $\phi_R(R(i))$, the i-th element of ϕ_R, is the probability that item $R(i)$ attracts the user, depending on the relevance and the diversity of the whole list. Finally, we sort the list by $\phi_R(R(i))$, and select the top-K items as the re-ranked list S for the user.

2) Probabilistic Approach: Inspired by the contextual bandit algorithms like LinUCB [40], [41], we further extend the deterministic approach and propose a probabilistic solution by involving exploration and randomization for diverse results.

Specifically, since stochastic sampling is non-differentiable, we adopt the reparameterization trick from the variational auto-encoder (VAE) [42] to make the back-propagation feasible. We separately estimate a mean and a standard deviation for the re-ranking score of each item,

$$\varphi_R = \text{MLP}_\varphi[H_R, \Delta_R]\,, \quad \Sigma_R = \text{MLP}_\Sigma[H_R, \Delta_R]\,, \tag{8}$$

where $\varphi_R \in \mathbb{R}^L$, $\Sigma_R \in \mathbb{R}^L$, and the i-th element denotes the mean and the standard deviation of the re-ranking score for the i-th item $R(i)$, $i = 1, \ldots, L$. In training, we transform the stochastic sampling into the following Eq.(9) by incorporating a standard normal random variable,

$$\phi_R(R(i)) = \varphi_{R(i)} + \xi_{R(i)}\Sigma_{R(i)}\,,\ \xi_{R(i)} \sim \mathcal{N}(0,1)\,, \tag{9}$$

where $\varphi_{R(i)}$ and $\Sigma_{R(i)}$ are trained via back-propagation. While at inference time, we use the upper confidence bound ($\varphi_{R(i)} + \Sigma_{R(i)}$) of the re-ranking score for item $R(i)$ to perform re-ranking,

$$U_R(R(i)) = \varphi_{R(i)} + \Sigma_{R(i)}\,. \tag{10}$$

The standard deviation $\Sigma_{R(i)}$ describes how confident the model is in the current estimated re-ranking scores and would like to encourage exploration, which is also user-specific. Likewise, we obtain the re-ranked list by selecting the top-K items sorted by the upper confidence bound of the re-ranking scores $U_R(R(i))$ to maximize the user's satisfaction.

E. Optimization

We minimize the cross entropy loss to push the model to score satisfactory items higher than unsatisfactory ones,

$$\mathcal{L} = -\sum_{l=1}^{n}\sum_{i=1}^{L}\{y_{R_l(i)}\log(\phi_{R_l}(R_l(i)))+(1-y_{R_l(i)})\log(1-\phi_{R_l}(R_l(i)))\}, \tag{11}$$

where $y_{R_l(i)} \in \{0, 1\}$ is the click indicator for the i-th item from list R_l with 1 denoting user clicks or conversions, and 0 denoting non-click, n is total number of user requests in the training set. To this end, the tradeoff between relevance and diversity can be jointly learned by RAPID in an end-to-end manner directly from user feedback.

IV. Experiments

Next, we empirically study the performance of RAPID towards answering the following research questions (RQs).

- (RQ1) How does RAPID perform compared with the state-of-the-art re-ranking models in terms of utility and diversity under various recommendation scenarios?
- (RQ2) How well does RAPID perform with different initial rankers?
- (RQ3) How does each designed component influence the performance of RAPID?
- (RQ4) How do the hyper-parameters (hidden size, maximum length of the behavior sequences) influence the performance of RAPID?
- (RQ5) How well does RAPID model personalized preference on diversity?

We first conduct semi-synthetic experiments on public datasets Taobao and MovieLens-20M. The semi-synthetic setup with a click model as the environment is widely applied [43], [44], which allows us to explore different settings and validate the advantage of our proposed model. The usage of the click model considers the possible position bias or selection bias in user behaviors, and could provide unbiased evaluation. We further conduct experiments on Huawei App Store dataset with real initial ranking lists and the corresponding clicks to highlight the effectiveness of RAPID in real-world applications.

A. Dataset Description

To evaluate the effectiveness of RAPID, we conduct experiments on two public datasets Taobao and MovieLens-20M, and an industrial dataset App Store.

1) Taobao: Taobao dataset[3] contains user purchase behaviors from Taobao, one of the biggest e-commerce platforms in China, with 987,994 users, 4,162,042 items, and 100,150,807 interactions, from November 25 to December 3, 2017. Data from the 1-2 days, 3-5 days, 6-8 days, and the last day are divided as user behavior history, initial ranker training set, re-ranking training set, and test set respectively. As the dataset contains 9,439 categories, we use Gaussian Mixture Models (GMMs) to cluster items into 5 topics as the item's topic coverage τ.

2) MovieLens-20M: MovieLens-20M dataset[4] contains 20,000,263 movie ratings between January 09, 1995, and March 31, 2015. We randomly divide the interactions into user behavior history, initial ranker training set, re-ranking training set, and test set in the ratio of 2:3:4:1 for each user, due to the long time span of the dataset. Given that all movies in MovieLens-20M belong to 20 genres, we use the normalized multi-hot genre vector as the item's topic coverage τ.

3) App Store: App Store dataset is collected from Huawei App Store, from June 1, 2022 to June 30, 2022, with 202,348,110 requests and 3,249 apps. Each app belongs to one of 23 different categories, so that the item's topic converge τ is a one-hot vector for the App Store dataset. The first 29 days are the training set, while the last day is the test set. The user behavior history is collected in real-time.

Notice that for Taobao and MovieLens, datasets are split into user behavior history, initial ranker training set, re-ranking training set, and test set. The initial ranker is trained on its initial ranker training set and provides initial ranking lists with predicted scores on the other data sets for training and testing the re-ranking models. Re-ranking models that consider the user behavior history like RAPID are trained on the re-ranking training set with user behavior history set as side information. Particular models that do not employ user behavior history are trained on the combined set of the re-ranking training data and the user behavior history data for fair comparison. All the models are tested on the same test set.

B. Experimental Settings

1) Click Feedback for Taobao & MovieLens: Taobao & MovieLens datasets do not have session and position information, so that cannot be directly adopted for re-ranking. To generate the corresponding click feedback for training and evaluation of the re-ranking models on these datasets, we involve a dependent click model (DCM) to simulate user clicks. DCM provides unbiased evaluation for a given list by considering multiple clicks and the dependencies between items, and is widely used for click simulation [43], [45]–[47].

In DCM, for each item v_k in list S at position k, the user is attracted and clicks on the item with an attraction probability $\bar{\phi}(v_k)$. After the click, she may be satisfied and leave with a termination probability $\bar{\epsilon}(k)$. If there is no click, the user will continue examining the next position. We follow the assumption in [37], [38], where the attraction probability is a combination of the item relevance and diversity, *i.e.*, $\tilde{\phi}(v_k) = \lambda\tilde{\alpha}(v_k) + (1-\lambda)\tilde{\rho}^\top\zeta(v_k)$, where λ is a hyper-parameter that measures the tradeoff between the item relevance $\tilde{\alpha}(v_k)$ and diversity $\tilde{\rho}^\top\zeta(v_k)$, $\zeta(v_k)$ is the coverage difference of item v_k compared to previously selected items, and $\tilde{\rho}$ is the weight parameter for each user. The parameters of DCM, *i.e.*, $\tilde{\alpha}$, $\tilde{\rho}$, and $\tilde{\epsilon}$, are estimated by maximizing the log-likelihood from the click logs follows [38], [45].

2) Evaluation Metrics: All re-ranking models are evaluated by commonly used utility and diversity metrics, where the utility is measured by the listwise total number of clicks

[3]https://tianchi.aliyun.com/dataset/dataDetail?dataId=649

[4]https://grouplens.org/datasets/movielens/20m/

TABLE II: Overall performance on public datasets.

(a) $\lambda = \mathbf{0.5}$

	Taobao								MovieLens-20M							
	click@5	ndcg@5	div@5	satis@5	click@10	ndcg@10	div@10	satis@10	click@5	ndcg@5	div@5	satis@5	click@10	ndcg@10	div@10	satis@10
Init	1.2358	0.3924	3.0807	0.2215	1.8793	0.4833	4.2193	0.1664	1.8412	0.4936	5.6425	0.3190	2.3379	0.5383	7.1999	0.2743
DLCM	1.7271	0.4742	3.2389	0.2972	2.3944	0.5788	4.3222	0.2043	2.0877	0.5406	6.4529	0.3543	2.6476	0.5892	8.0529	0.3048
PRM	1.7900	0.4828	3.1775	0.3064	2.4485	0.5876	4.2697	0.2079	2.0911	0.5267	6.2818	0.3550	2.6560	0.5775	7.8888	0.3057
SetRank	1.6935	0.4553	3.1826	0.2925	2.2972	0.5629	4.2621	0.1963	2.0630	0.5185	6.2412	0.3511	2.6291	0.5692	7.8875	0.3032
SRGA	1.7801	0.5013	3.1874	0.3052	2.4890	0.6109	4.2915	0.2117	2.0814	0.5228	6.3382	0.3536	2.6448	0.5737	7.9638	0.3046
MMR	1.2560	0.4100	3.2464	0.2247	1.8938	0.5048	4.2611	0.1674	1.8575	0.4857	5.8207	0.3214	2.3455	0.5280	7.3301	0.2750
DPP	1.3326	0.4259	**4.1645**	0.2377	1.8732	0.5023	**4.5851**	0.1645	1.9458	0.5065	6.4568	0.3342	2.4378	0.5480	**8.3408**	0.2846
DESA	1.7479	0.4745	3.1903	0.3002	2.4121	0.5794	4.2841	0.2055	2.0707	0.5253	6.2853	0.3521	2.6374	0.5754	7.9044	0.3039
SSD	1.1988	0.3296	3.1513	0.2156	1.8433	0.4214	4.2822	0.1638	1.8135	0.3888	5.7663	0.3151	2.3127	0.4397	7.3705	0.2717
adpMMR	1.2408	0.3962	3.1179	0.2221	1.8802	0.4852	4.2471	0.1663	1.844	0.4924	5.6663	0.3194	2.3385	0.5358	7.2139	0.2744
PD-GAN	1.2478	0.4072	3.1092	0.2235	1.8919	0.5009	4.3134	0.1674	1.8601	0.4901	5.9314	0.3287	2.6307	0.5640	8.0515	0.2806
RAPID-det	1.8323	0.5082	3.2356	0.3134	**2.5551***	**0.6214***	4.327	**0.2167***	2.0981	0.5423	6.4644	0.3558	2.6591	0.5899	8.0709	0.3059
RAPID-pro	**1.8358***	**0.5102***	3.2265	**0.3135***	2.5110	0.6149	4.3130	0.2127	**2.1071***	**0.5442***	**6.5210***	**0.3570***	**2.6648***	**0.5928***	8.1261	**0.3064***

(b) $\lambda = \mathbf{0.9}$

	Taobao								MovieLens-20M							
	click@5	ndcg@5	div@5	satis@5	click@10	ndcg@10	div@10	satis@10	click@5	ndcg@5	div@5	satis@5	click@10	ndcg@10	div@10	satis@10
Init	1.2648	0.3248	3.0807	0.2256	2.1741	0.4254	4.2193	0.1928	1.7972	0.4067	5.6412	0.3103	2.4889	0.4418	7.1997	0.2871
DLCM	2.5170	0.6640	3.1926	0.4048	3.4267	0.7557	4.2718	0.2774	2.2393	0.4871	5.8975	0.3736	3.0700	0.5324	7.5226	0.3428
PRM	2.5965	0.6918	3.1957	0.4147	3.4878	0.7798	4.2704	0.2809	2.2408	0.4874	5.8957	0.3737	3.0715	0.5325	7.5191	0.3430
SetRank	2.3785	0.6152	3.1870	0.3893	3.3416	0.7159	4.2706	0.2731	2.2401	0.4820	5.8967	0.3738	3.0700	0.5273	7.5221	0.3428
SRGA	2.5803	0.6949	3.1925	0.4132	3.5103	0.7873	4.2719	0.2831	2.2425	0.4856	5.8992	0.3740	3.0716	0.5304	7.5195	0.3429
MMR	1.2636	0.3264	3.2439	0.2256	2.1824	0.4302	4.2708	0.1937	1.8005	0.4060	5.7282	0.3108	2.4905	0.4415	7.2514	0.2873
DPP	1.2109	0.3188	**4.1641**	0.2174	2.0838	0.4164	**4.5847**	0.1858	1.8077	0.4052	**6.4558**	0.3119	2.4959	0.4406	**8.3398**	0.2879
DESA	2.5292	0.6732	3.1915	0.4060	3.4341	0.7636	4.2724	0.2778	1.9935	0.4357	5.9985	0.3391	2.7671	0.4775	7.6097	0.3144
SSD	1.2174	0.3059	3.1434	0.2183	2.1156	0.4071	4.2766	0.1887	1.6715	0.3544	5.6804	0.2915	2.3710	0.3954	7.2746	0.2750
adpMMR	1.2649	0.3255	3.1179	0.2256	2.1739	0.4261	4.2271	0.1928	1.7982	0.4053	5.6663	0.3105	2.4887	0.4409	7.2139	0.2871
PD-GAN	1.2814	0.3338	3.0978	0.2280	2.2038	0.4350	4.3162	0.1953	1.8090	0.4098	5.6617	0.3121	3.0015	0.4792	8.2334	0.2743
RAPID-det	2.6571	0.7077	3.1899	0.4227	3.5668	0.7969	4.2726	**0.2862***	2.2433	0.4878	5.9062	0.3741	3.0752	0.5336	7.5222	0.3431
RAPID-pro	**2.6575***	**0.7084***	3.1966	**0.4228***	**3.5680***	**0.7976***	4.2770	**0.2862***	**2.2457**	**0.4882**	5.9103	**0.3745***	**3.0761***	**0.5332**	7.5240	**0.3432***

(c) $\lambda = \mathbf{1.0}$

	Taobao								MovieLens-20M							
	click@5	ndcg@5	div@5	satis@5	click@10	ndcg@10	div@10	satis@10	click@5	ndcg@5	div@5	satis@5	click@10	ndcg@10	div@10	satis@10
Init	1.2721	0.3132	3.0807	0.2267	2.2478	0.4156	4.2193	0.1993	1.7862	0.3897	5.6412	0.3085	2.5265	0.4244	7.1992	0.2905
DLCM	2.7192	0.7237	3.1921	0.4298	3.6927	0.8081	4.2698	0.2950	2.2833	0.5006	5.7944	0.3797	3.1765	0.5410	7.4217	0.3524
PRM	2.7903	0.7483	3.1932	0.4381	3.7401	0.8289	4.2694	0.2975	2.2828	0.5008	5.8059	0.3797	3.1759	0.5417	7.4360	0.3524
SetRank	2.5298	0.6449	3.1922	0.4100	3.6154	0.7465	4.2759	0.2925	2.2821	0.4981	5.8082	0.3793	3.1762	0.5396	7.4277	0.3522
SRGA	2.7713	0.7466	3.1924	0.4366	3.7669	0.8330	4.2723	0.3000	2.2831	0.5010	5.8046	0.3794	3.1761	0.5423	7.4262	0.3522
MMR	1.2635	0.3127	3.2440	0.2256	2.2521	0.4172	4.2708	0.2000	1.7862	0.3902	5.6414	0.3085	2.5266	0.4248	7.1996	0.2905
DPP	1.1792	0.2990	**4.1639**	0.2121	2.1347	0.4022	**4.5848**	0.1909	1.7733	0.3890	**6.4559**	0.3067	2.5109	0.4228	**8.3403**	0.2890
DESA	2.7908	0.7471	3.1963	0.4382	3.7406	0.8282	4.2730	0.2976	2.0387	0.4423	5.8634	0.3454	2.8748	0.4823	7.4509	0.3243
SSD	1.2284	0.3098	3.1487	0.2199	2.1909	0.4106	4.2786	0.1954	1.6794	0.3641	5.7220	0.2924	2.4268	0.4003	7.3414	0.2803
adpMMR	1.2709	0.3142	3.1179	0.2265	2.2473	0.4156	4.2271	0.1993	1.7867	0.3895	5.6663	0.3086	2.5262	0.4252	7.2139	0.2905
PD-GAN	1.2815	0.3262	3.1017	0.2284	2.2358	0.4239	4.3338	0.1981	1.8004	0.3974	5.6305	0.3106	3.0967	0.4635	8.2268	0.2810
RAPID-det	2.8209	0.7580	3.1945	0.4429	**3.8455***	0.8474	4.2723	**0.3053***	2.2872	0.5021	5.8049	0.3799	3.1822	0.5425	7.4341	0.3525
RAPID-pro	**2.8579***	**0.7652***	3.1973	**0.4470***	3.8338	**0.8475***	4.2743	0.3036	**2.2878***	**0.5027**	5.8101	**0.3800***	**3.1822***	**0.5436**	7.4384	**0.3526***

$*$ denotes statistically significant improvement (measured by t-test with p-value$<$0.05) over all baselines.

click@k, the popular learning-to-rank metric *ndcg@k* [48], the overall user satisfaction *satis@k*, and the total amount of income for the platform *rev@k*, while the diversity is measured by the topic coverage *div@k*. The *click@k* and *rev@k* are the major concerns for industrial recommender systems [44]. For all the below metrics, we report $k = 5, 10$.

Click@k. The total number of clicks is defined by $click@k = \frac{1}{n}\sum_{l=1}^{n}\sum_{i=1}^{k} y_l(v_i)$, where $y_l(v_i) \in \{0, 1\}$ denotes the click of the i-th item for request l.

Satis@k. For Taobao and MovieLens-20M, the average user satisfaction [46], [47], given by DCM, is defined by $satis@k = 1 - \frac{1}{n}\sum_{l=1}^{n}\prod_{i=1}^{k}\left(1 - \tilde{\epsilon}_l(i)\tilde{\phi}_l(v_i)\right)$, where $\tilde{\epsilon}_l(i)$ and $\tilde{\phi}_l(v_i)$ are termination probability at position k and the attraction probability of item v_i for request l, respectively.

Rev@k. For App Store dataset, since the evaluation does not depend on the click model and the objective of the platform is to optimize the total revenue, we choose *rev@k* as the main utility metric, defined by $rev@k = \frac{1}{n}\sum_{l=1}^{n}\sum_{i=1}^{k} b_l(v_i)y_l(v_i)$, where $b_l(v_i)$ is the given bid price of item v_i for request l.

Div@k. The topic coverage *div@k* denotes the expected number of covered topics for the top-k items, which is widely adopted in existing studies [37], [39], [49], [50], *i.e.*, $div@k = \frac{1}{n}\sum_{l=1}^{n}\sum_{j} c_{l,j}(S_{1:k})$, where n is the total number of requests, and $c_{l,j}(S_{1:k})$ is the coverage of topic j for the top-k items

in the re-ranked list $S_{1:k}$ for request l.

3) Initial Rankers and Baselines: We select three widely adopted ranking algorithms, including the representative deep ranking model DIN [51], SVMRank [52], and LambdaMART [53], to generate the initial ranking lists, which use pointwise, pairwise, and listwise loss, respectively.

To demonstrate the effectiveness, we compare the proposed RAPID with the following state-of-the-art re-ranking models, including relevance-oriented re-ranking models (DLCM, PRM, SetRank, and SRGA), diversity-aware re-ranking models (MMR, DPP, DESA, and SSD), and personalized diversity models (adpMMR and PD-GAN)[5]:

DLCM [7] is the re-ranking model that first applies GRU, which encodes top-ranking items to learn a local context embedding. **PRM** [2] employs a self-attention mechanism to explicitly model the cross-item interactions. **SetRank** [20] adopts a stack of (induced) multi-head self-attention blocks to learn a permutation-invariant ranking model. **SRGA** [19] improves the attention structure by considering the impact of unidirectivity and locality. **MMR** [24] diversifies the initial ranking lists by greedily adding items with maximal marginal relevance. **DPP** [10], [28] uses Determinantal Point Process to generate relevant and diverse recommendations. **DESA** [9] is a state-of-the-art diversity-aware method that jointly estimates diversity and relevance with a self-attention structure and a pairwise loss. **SSD** [29] improves the diversity by maximizing the volume spanned by the constructed trajectory tensor with sliding windows. **AdpMMR** [32] is a personalized diversity model that computes users' propensity toward diversity by feature entropy. **PD-GAN** [12] is a personalized diversity model with a personalized DPP kernel matrix designed for the ranking stage.

C. Reproducibility

The implementation of our model is publicly available[6]. We adopt Adam [54] as the optimizer. The maximum length of user behavior sequences for each topic is set to 5, and the maximum length of the initial list is set to 20. The learning rate is selected from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, the batch size from $\{256, 512, 1024\}$, and the hidden size from $\{8, 16, 32, 64\}$ with grid search. To ensure fair comparison, we also fine-tune all baselines to achieve their best performance.

D. (RQ1) Overall Performance

We compare RAPID with several state-of-the-art baselines, on two public benchmark datasets and an industrial dataset. RAPID with deterministic approach and probabilistic approach are denoted by RAPID-det and RAPID-pro, respectively.

[5]We did not include the results of the personalized diversity method LPF [30], because with careful parameter tuning, the utility metrics are still notably worse than the initial ranking. Possible reasons are (i) The matrix factorization model that LPF based on is limited when the dataset is extremely sparse such as Taobao or MovieLens-20M, and does not integrate the content features. If the quality of latent factors is inferior, the estimated utility may be imprecise. (ii) LFP is a two-phase model, the optimization directions of the two phases may not be consistent. We did not compare PS [31] in our experiments since the personality values are unobtainable for the datasets.

[6]Our code is available at https://github.com/wwliu555/RAPID

1) Benchmark Datasets: To study the performance of the re-ranking models under different recommendation scenarios, we set the relevance-diversity tradeoff parameter λ of the click model to $0.5, 0.9$, and 1.0, where the importance of diversity in motivating clicks decreases accordingly. The setting $\lambda = 0.5$ indicates relevance and diversity are roughly of the same importance, like news feed recommendation scenario [55], [56] where user clicks also depend a lot on diversity. For $\lambda = 1$, it represents the scenario that user clicks are simply motivated by relevance, like ads recommendation [57]. We select the deep ranking model DIN as our initial ranker. The overall performance on Taobao and MovieLens-20M is reported in Table II, from which we have several observations.

First, the initial rankers usually do not perform well in both utility and diversity compared to the performance of the re-ranking models. As suggested in Table II, we observe that the initial ranker (Init) has the lowest utility and diversity. All purely relevance-oriented re-ranking models (including DLCM, PRM, SetRank, and SRGA) improve the initial ranker in both utility and diversity by a large margin, due to the incorporation of the listwise context. In general, PRM and SRGA perform best amongst all the baselines, but the results of SRGA vary under different settings and are not as good on the later industrial dataset. Whereas diversity-aware re-ranking models usually have a great improvement in diversity, the increase in utility is only marginal. DESA generally can achieve better diversity and utility, but its performance is not so stable under different settings. Although adpMMR and PD-GAN consider the personalized diversity, their performance gain in utility over the initial ranker is not so significant. It is because adpMMR simply employs the entropy of user features to determine the degree of diversity, which is rule-based and non-learnable. As for PD-GAN, the personalized diversity is described by the number of topics favored by the user in PD-GAN, which has limited expressive power. Moreover, PD-GAN is designed for the ranking stage and estimates the relevance for each candidate item independently, ignoring the listwise context.

Second, the relevance-diversity tradeoff exists widely. Considering the diversity-aware models like DPP, though they achieve higher diversity than the initial ranker, the utility-based metrics are much inferior to the relevance-oriented models. The concepts of relevance and diversity are inherently conflicting to some extent. To obtain ideal diversity, one could simply divide the recommendation equally for each topic. Contrastingly, the main goal of learning relevance is to break the equality and model users' personalized tastes over different topics. DPP is specially designed to promote diversity by selecting the most dissimilar subset of the candidate items, but is at a great sacrifice of relevance. In particular, DPP attains the largest diversity at a decrease of 37.8% in *click@5* on Taobao for $\lambda = 0.5$ compared to RAPID-pro, which is undesirable. RAPID, however, is not designed to only maximize diversity, but to optimize the overall utility by diversifying the results according to individual user's preference.

Third, RAPID outperforms relevance-oriented re-ranking

TABLE III: Overall performance on App Store dataset.

	click@5	ndcg@5	div@5	rev@5	click@10	ndcg@10	div@10	rev@10
Init	0.8618	0.4353	3.4678	3.8806	0.9981	0.4733	4.3987	4.5659
DLCM	0.9055	0.5871	3.5111	4.4447	1.0357	0.6238	4.4112	4.9221
PRM	0.9561	0.6258	3.4469	4.4897	1.0539	0.6528	4.3989	4.9379
SetRank	0.9364	0.6040	3.5299	4.4809	1.0424	0.6336	4.4110	4.9377
SRGA	0.9133	0.6058	3.4502	4.4229	1.0399	0.6492	4.3907	4.8968
MMR	0.9177	0.5555	3.4446	4.2867	1.0571	0.6297	4.3813	4.5387
DPP	0.8959	0.5290	**3.8925**	4.0006	0.9916	0.6026	**4.5752**	4.4658
DESA	0.9320	0.6035	3.5558	4.4608	1.0378	0.6331	4.4057	4.9177
SSD	0.9071	0.5565	3.4569	4.4048	1.0275	0.6110	4.4034	4.5857
adpMMR	0.9284	0.5357	3.4626	4.3621	1.0496	0.6134	4.3876	4.5659
PD-GAN	0.9081	0.5837	3.4632	4.3810	1.0336	0.6297	4.4048	4.5764
RAPID-det	0.9860	0.6595	3.6481	4.5501	1.0700	0.6585	4.4692	4.9746
RAPID-pro	**1.0002***	**0.6693***	3.6509	**4.5523***	**1.0786***	**0.6825***	4.4858	**4.9909***
impv%	4.61%	6.95%	5.92%	2.06%	2.34%	4.55%	1.98%	1.07%

$*$ denotes statistically significant improvement (p-value$<$0.05) over PRM, which achieves the highest *rev@k* among baselines.

TABLE IV: Comparison on different initial ranking lists.

	SVMRank				LambdaMart			
	Taobao		MovieLens-20M		Taobao		MovieLens-20M	
	click@10	div@10	click@10	div@10	click@10	div@10	click@10	div@10
Init	2.0854	4.2271	2.0423	6.6047	2.0446	4.1870	2.0892	6.4269
DLCM	3.5474	4.2596	2.8468	6.7615	3.6150	4.2597	2.8548	7.0649
PRM	3.6053	4.2630	2.8481	6.7952	3.6200	4.2515	2.8566	7.0706
SetRank	3.4469	4.2561	2.8478	6.7670	3.4261	4.2629	2.8550	7.0629
SRGA	3.6326	4.2624	2.8490	6.7790	3.6380	4.2555	2.8569	7.0701
MMR	2.1009	4.2289	2.0427	6.6171	2.0578	4.1863	2.0893	6.4287
DPP	2.1979	**4.6167**	2.1547	7.3142	2.1234	**4.6654**	2.2806	**7.8096**
DESA	3.5519	4.2607	2.7641	6.8276	3.1601	4.2660	2.5849	7.1403
SSD	2.0692	4.2499	2.0204	6.6532	2.0954	4.3065	2.0631	6.7209
adpMMR	2.0954	4.228	2.0436	6.6072	2.0546	4.1873	2.0893	6.4272
PD-GAN	2.0993	4.3272	2.5034	**7.5758**	2.1054	4.2778	2.5861	7.4944
RAPID-det	**3.7049**	4.2584	2.8504	6.7719	3.7113	4.2583	2.8598	7.0788
RAPID-pro	3.6791	4.2656	**2.8522**	6.8051	**3.7118**	4.2613	**2.8613**	7.0891

models in different metrics under different settings. The results in Table II show that RAPID performs significantly better than other baselines. Compared to the relevance-oriented re-ranking models, RAPID consistently yields better performance w.r.t. both utility and diversity metrics under all settings. For instance, RAPID-pro surpasses PRM by 2.55% in *click@10* and 1.01% in *div@10* on Taobao, and 0.33% in *click@10* and 3.01% in *div@10* on MovieLens-20M at $\lambda = 0.5$. We attribute this improvement to RAPID automatically extracting users' personalized diversity preferences from their behavior history, and performing re-ranking with customized diversification. As such, for one thing, the relevance-based metrics are enhanced by providing more diverse options that fit the user's interests instead of duplicate items, where the user usually clicks at most one of them. For another, different users have different intents for diversity, providing diverse items for users with diverse interests improves the overall diversity metrics. Moreover, RAPID intelligently learns to fuse relevance and diversity in an end-to-end manner. When we increase λ from 0.5 to 1, RAPID is capable of automatically putting less emphasis on diversity in general, *e.g.*, the improvements of RAPID-pro in *div@10* on Taobao over PRM are reduced from 1.01%, 0.15%, to 0.11%, and improvements in *click@10* are relatively stable, *i.e.*, 2.55%, 2.30%, 2.51%, showing that RAPID can well adapt to different recommendation scenarios and generate the most satisfactory re-ranking results. Furthermore, considering the different output approaches, RAPID-pro generally achieves better results than RAPID-det due to the benefit of encouraging exploration in re-ranking.

2) Industrial Datasets: We further test RAPID on Huawei App Store and evaluate RAPID directly by real-world click-through data, without the click model. The objective of the App Store is to maximize the total revenue. Table III presents the results of RAPID and the state-of-the-art baselines. The initial ranking list (Init) is generated by the App Store's ranking strategy currently running on the platform. Similar observations as on the public datasets can be observed for the App Store dataset. The relevance-oriented re-ranking models offer better re-ranked performance in both utility and diversity compared to the initial ranking list. The diversity-aware models like DPP exhibit a great rise in diversity, yet with a large fall in utility. In particular, RAPID attains a better performance under various metrics. RAPID improves the strongest baseline PRM by 2.06% in *rev@5* and 1.07% in *rev@10*, while in the meantime providing a more diverse re-ranking result (improved by 5.92% and 1.98% in *div@5* and *div@10*, respectively). Such an improvement in *rev@k* corresponds to a significant increase in income for the platform. This is because RAPID captures users' personalized preferences over all topics by modeling the intra- and inter-topic interactions from the users' behavior history, which benefits the re-ranking performance. In addition, RAPID directly estimates the relevance and the diversity from user clicks in a unified end-to-end framework so that is capable of jointly learning the combination of relevance and diversity, instead of explicitly enforcing diversity by the tradeoff parameter. Since RAPID-pro generates better results in general, in the hereafter, we adopt RAPID-pro as RAPID unless otherwise specified.

E. In-depth Analysis

1) (RQ2): Different Initial Rankers: As various recommendation scenarios and platforms may adopt different initial rankers, we explore whether our model can be generalized to other initial ranking algorithms. In particular, we further implement SVMRank and LambdaMART as the initial ranker for comparison. Due to the page limit of the paper, we only present a utility-based metric (*click@10*) and a diversity-based metric (*div@10*) for the case of $\lambda = 0.9$ in Table IV. Although SVMRank and LambdaMART perform worse than DIN in terms of initial ranking performance, the trend of re-ranking models on SVMRank and LambdaMART is similar to that on DIN. The tradeoff between utility and diversity also exists, and DPP performs poorly on utility as it focuses too much on diversity. Nevertheless, RAPID can still significantly improve utility and maintain better performance in diversity than those relevance-oriented re-ranking algorithms.

2) (RQ3) Ablation Analysis: Four variants of RAPID are devised to investigate the effectiveness of the designed component: **RAPID-RNN** removes the personalized diversity estimator. **RAPID-mean** replaces LSTM in the personalized diversity estimator with a simple mean aggregation of the item embedding for each topic. **RAPID-det** replaces the probabilistic output approach with the deterministic output

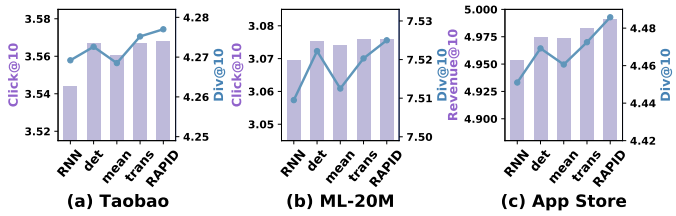

Fig. 3: Performance of different variants of RAPID.

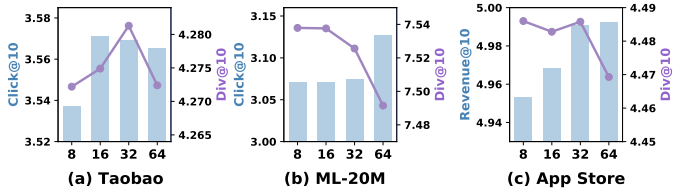

Fig. 4: RAPID with different hidden sizes.

approach. **RAPID-trans** replaces the Bi-LSTM in the listwise relevance estimator by a transformer structure.

The comparison of these four variants and the original RAPID is shown in Figure 3. For RAPID-RNN, after removing the module of personalized diversity estimator, both *div@10* and *click@10* have dropped significantly, demonstrating that personalized diversification in re-ranking substantially enhances the re-ranking performance. The removal of the probabilistic output approach in RAPID-det introduces a decline in *div@10*. It hence verifies the upper confidence bound in the probabilistic approach can encourage more diverse re-ranking results while maintaining or improving the utility. Compared with RAPID, RAPID-mean has a decrease in diversity, which indicates the LSTM of the personalized diversity estimator captures the intra-topic interactions and diverse interests in user history. By changing Bi-LSTM to a transformer, RAPID-trans obtains similar *click@10* to RAPID, with slightly lower diversity. Possible reasons may be that the transformer relies on additional position embeddings to extract sequential information, which is relatively hard to train compared to the RNN-like structure that inherently takes the order into account.

3) (RQ4) Hyper-parameter Study: Practically, we notice that the hyper-parameters, especially the hidden size q_h and the length of the behavioral sequences D, influence the final results. Thus, we conduct grid-search experiments to get a comprehensive understanding of how those hyper-parameters affect RAPID's performances. For the two public datasets, we report the results with $\lambda = 0.9$ and DIN as the initial ranker.

As illustrated in Figure 4, for the hidden size, *div@10* decreases while *click@10* increases as the hidden size grows on MovieLens-20M and App Store, which also demonstrates the existence of relevance-diversity tradeoff. While on Taobao, RAPID suffers from a small dimension (*e.g.*, 8) due to its limited fitting capability. Yet a large dimension (*e.g.*, 64) may cause overfitting and also degrades the performance.

As for the length of the behavioral sequences, since the change in behavior history also affects the performance of the initial ranker DIN, we only conduct experiments on App Store dataset to avoid the influence of the initial ranker. As shown in Table V, RAPID generally performs best at $D = 5$. Naturally, too few behaviors ($D = 3$) may not be able to carry enough information for learning the users' diversity preferences so that both the utility and the diversity are lower. When $D = 10$, though diversity remains high, the utility-based metrics decrease. It may be because long behaviors introduce some noise and affect the performance.

TABLE V: RAPID with different maximum lengths of behavior sequences on App Store dataset.

	click@5	ndcg@5	div@5	rev@5	click@10	ndcg@10	div@10	rev@10
RAPID-3	0.9956	0.6604	3.6477	4.5521	1.0707	0.6793	4.4819	4.9834
RAPID-5	1.0002	**0.6693**	3.6509	**4.5523**	**1.0786**	**0.6825**	**4.4858**	**4.9909**
RAPID-10	**1.0013**	0.6669	**3.6511**	4.5439	1.0700	0.6786	4.4809	4.9875

TABLE VI: Training and inference time comparison.

	Taobao			MovieLens-20M			App Store		
	Train-all (min)	Train-b (ms)	Test-b (ms)	Train-all (min)	Train-b (ms)	Test-b (ms)	Train-all (min)	Train-b (ms)	Test-b (ms)
PRM	35	13.0	9.6	118	20.7	17.5	98	40.7	32.3
DESA	52	16.3	9.8	165	21.7	17.9	137	42.2	33.0
RAPID	32	20.4	10.9	114	26.7	18.4	106	45.6	35.3

4) Efficiency Study: In order to quantitatively analyze the time complexity, we compare RAPID with two state-of-the-art neural baselines and record their total training time until convergence (referred to as *train-all*) and average training and inference time for each batch (referred to as *train-b* and *test-b*, respectively), summarized in Table VI. Experiments are conducted on NVIDIA 3080 GPU with 10G memory for public datasets, and on Tesla V100 with 32G memory for the App Store dataset. We observe that RAPID converges faster than DESA and yields a low total training time. Its training and inference time per batch is comparable to other baselines. Although RAPID may be slightly slower due to the modeling of the user behavior history, this is worthwhile since the utility improvement is considerable (about 2%). Moreover, the inference time of RAPID can meet the common response latency requirement of industrial recommender systems, within 50 ms [58].

F. (RQ5) Case Study

In this section, we aim to study whether RAPID is capable of actually learning the users' personalized preferences towards diversity. We therefore select two users from the MovieLens-20M dataset, one with diverse interests and one with relatively focused interests, and visualize the genre distribution of their interested movies in history and top-ranked movies produced by RAPID.

Figure 5 presents a multi-interest user 1, whose interested movie covers a wide range of genres. We observe that RAPID fully explores various combinations of genres that the user is interested in and appropriately recommends new genres that the user would like to explore, such as War. The interests of user 2 in Figure 5 are relatively homogeneous – almost all the genres favored by user 2 are drama. RAPID captures this personalized diversity preference well. It explores combinations of genres mostly with drama. Besides, user 2 also reveals that RAPID is aware of users' preferences for certain combination types, such as drama and romance. Therefore,

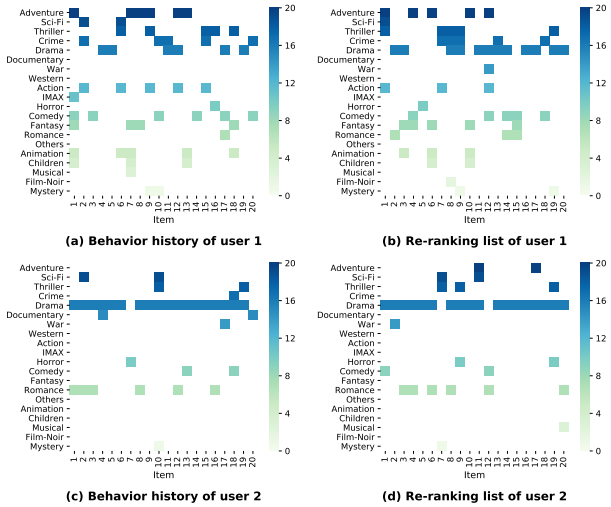

Fig. 5: The genres of items for different types of users.

we can conclude that RAPID is capable of performing re-ranking with personalized diversification in accordance with the personal preference over different topics.

V. Theoretical Analysis

A. Efficacy Analysis

The goal of RAPID is to maximize the overall utility, which is a compound of the relevance and the diversity. In this section, we prove that the performance of RAPID is guaranteed to produce a near-optimal utility. To make the analysis manageable and get a more intuitive understanding of the proposed algorithm, we assume that the user's click feedback follows a linear dependent click model (DCM) [45] as in Section IV-B1, where the click probability is a linear combination of relevance and diversity and the diversity coefficient is user-specific. Then the re-ranking function of RAPID can be simplified to $\phi_R = \hat{\beta}^\top \mathcal{R} + \hat{b}^\top \mathcal{T} d_R$ by replacing the complicated network structure like Bi-LSTM with a linear form. Here, $\mathcal{R} \in \mathbb{R}^{(q_u+q_v+m)\times L}$ is the feature matrix of the initial ranking list , $\mathcal{T} \in \mathbb{R}^{(q_u+q_v)D\times m}$ is the personalized behavior sequences, $d_R \in \mathbb{R}^{m\times L}$ represents the marginal diversity of the list R in Eq.(5), and $\hat{\beta} \in \mathbb{R}^{q_u+q_v+m}$ and $\hat{b} \in \mathbb{R}^{(q_u+q_v)D}$ are the learned model parameters. Denote by β_*, b_* the unknown ground truth parameters.

In particular, the first term $\hat{\beta}^\top \mathcal{R}$ is the listwise relevance, and the second term $\hat{b}^\top \mathcal{T} d_R$ is the personalized diversity. The preference distribution $\hat{\theta}$ in Eq. (3) is now represented by $\hat{\theta}^\top = \hat{b}^\top \mathcal{T}$ which is user-specific, so we have $\phi_R = \hat{\beta}^\top \mathcal{R} + \hat{\theta}^\top d_R$. For simplicity, let $\hat{\omega} = [\hat{\beta}^\top, \hat{\theta}^\top]^\top$, $\eta = [\mathcal{R}^\top, d_R^\top]^\top$, and thus ϕ_R can be replaced by $\phi_R = \hat{\omega}^\top \eta$. Likewise, we define the underlying attraction probability $\bar{\phi}_R = \omega_* \eta$. We follow the assumption in [47] that the order of the termination probabilities is known. Without loss of generality, let $\bar{\epsilon}(1) \geq \ldots \geq \bar{\epsilon}(K)$. In the later analysis, we focus on the list of top-K items S, which is selected from the input initial list R. The γ-scaled n step regret is

$$\mathcal{G}_\gamma(n) = \mathbb{E}\Big[\sum\nolimits_{u=1}^{n} \big(f(S_u^*, \bar{\epsilon}_u, \bar{\phi}_{S_u^*}) - f(S_u, \bar{\epsilon}_u, \bar{\phi}_{S_u})/\gamma\big)\Big], \quad (12)$$

where $f(S, \epsilon, \phi) = 1 - \prod_{k=1}^{K}(1 - \epsilon(k)\phi_S(v_k))$ is the satisfaction (utility) function for DCM, $\epsilon(k)$ is the termination probability at position k, $S_u^* = [v_{1,*}, \ldots, v_{K,*}]$ is the optimal re-ranking lists, $\gamma = (1 - \frac{1}{e}) \max\left\{\frac{1}{K}, 1 - \frac{2}{K-1}\bar{\phi}_{\max}\right\}$ is the approximation ratio of a greedy method [37], with $\bar{\phi}_{\max} = \max_S \max_{v\in S} \bar{\phi}(v)$ denoting the maximum attraction probability. Let $p_v = \max_{u=1:n} \max_{k=1:K} \bar{\epsilon}_u(k) - \bar{\epsilon}_u(k+1)$ be the maximal difference of the termination probability of two consecutive positions. Frequently used symbols are listed in Table I. Our main result of the bounded γ-scaled regret of RAPID is given by Theorem 5.1,

Theorem 5.1: For any $\sigma > 0$, $\|\omega_*\|_2 \leq 1$ and any

$$s \geq \frac{1}{\sigma}\sqrt{q_0 \log\left(1 + \frac{nK}{q_0\sigma^2}\right) + 2\log(n)} + \|\omega_*\|_2\,,$$

we have

$$\mathcal{G}_\gamma(n) \leq \frac{2p_v s K^2}{\gamma}\sqrt{\frac{q_0 n \log(1 + \frac{nK}{q_0\sigma^2})}{\log(1 + \frac{1}{\sigma^2})}} + 1\,,$$

where $q_0 = q_u + q_v + 2m$. The regret has a $\tilde{O}(q_0\sqrt{n})$ regret bound, where the $\tilde{O}$ notation hides logarithmic factors.

The result has the following characteristics: (1) The theorem states a gap-free bound, where the factor $\sqrt{n}$ is considered *near-optimal* w.r.t. n, matching the lower bound up to polylogarithmic factors [37]. This shows that our algorithm is capable of achieving satisfying long-term overall utility (equivalently, minimizing the regret); (2) It depends linearly on the dimension of the user feature q_u, item feature q_v, and the number of topics m, which is standard in linear bandits [59]; (3) This is the first time that we discuss the regret bound of an algorithm with personalized diversification for multiple clicks, which generalizes the previous cascade model assumption [37], [38] to involve multiple clicks and personalization; (4) Existing re-ranking methods, on the contrary, either *solely* optimize relevance or ignore the *personalization* in diversity, and thereby could not obtain same satisfying utility.

Proof. To begin with, we bound the one-step regret for a user u, $u = 1, \ldots, n$. With a slight abuse of notation, here, u represents a request made by the user at the corresponding time step. Let $M_u = I_{q_0} + \sum_{j=1}^{u}\sum_{i=1}^{K} \eta_{S_j,i}\eta_{S_j,i}^\top$, where $\eta_{S_j,i}$ is the input of the i-th item of list S_j. We derive the high-probability upper confidence bound U_{S_u}, and its high-probability lower confidence bound L_u of $\bar{\phi}_{S_u}$, from Lemma 1 in [60], for a re-ranked list S_u as

$$U_{S_u} = \text{Proj}_{[0,1]}\left[\hat{\omega}^\top \eta_{S_u} + s\sqrt{\eta_{S_u}^\top M_u^{-1} \eta_{S_u}}\right],$$
$$L_{S_u} = \text{Proj}_{[0,1]}\left[\hat{\omega}^\top \eta_{S_u} - s\sqrt{\eta_{S_u}^\top M_u^{-1} \eta_{S_u}}\right].$$

Note that $\text{Proj}_{[0,1]}$ projects each real number onto interval $[0, 1]$, and $s > 0$ controls the degree of the exploration/uncertainty. We further denote the confidence radius $\rho_u(v_k) = s\sqrt{\eta_{S_u}(k)^\top M_u^{-1} \eta_{S_u}(k)}$ for list S_u.

We define a *good event* $\mathcal{E}_u = \{L_{S_u} \le \bar{\phi}_{S_u} \le U_{S_u}, \forall S_u\}$ that the attraction probability is bounded by the upper and lower bound, and let $\bar{\mathcal{E}}_u$ be the complement of $\mathcal{E}_u$. Then

$$\mathbb{E}\Big[f(S_u^*, \bar{\epsilon}_u, \bar{\phi}_{S_u}) - f(S_u, \bar{\epsilon}_u, \bar{\phi}_{S_u})/\gamma\Big] \le P(\bar{\mathcal{E}}_{u-1}) + P(\mathcal{E}_{u-1})\mathbb{E}[f(S_u^*, \bar{\epsilon}_u, \bar{\phi}_{S_u}) - f(S_u, \bar{\epsilon}_u, \bar{\phi}_{S_u})/\gamma|\mathcal{E}_{u-1}], \quad (13)$$

which is simply due to $f(S_u^*, \bar{\epsilon}_u, \bar{\phi}_{S_u}) - f(S_u, \bar{\epsilon}_u, \bar{\phi}_{S_u})/\gamma \le 1$.

As described in Section III-D2, the re-ranked list of RAPID is obtained by selecting the top-K items sorted by the upper confidence bound. Therefore, due to $L_{S_u} \le \bar{\phi}_{S_u} \le U_{S_u}$ for any S_u under event $\mathcal{E}_u$, and $f(S, \epsilon, \phi)$ is a non-decreasing function w.r.t. ϕ, we have $f(S_u^*, \bar{\epsilon}_u, \bar{\phi}_{S_u^*}) \le f(S_u^*, \bar{\epsilon}_u, U_{S_u^*})$. On the other hand, since S_u is computed based on a γ-approximation algorithm, by definition $f(S_u^*, \bar{\epsilon}_u, U_{S_u^*}) \le f(S_u, \bar{\epsilon}_u, U_{S_u})/\gamma$. Combining the above two inequalities, we have $f(S_u^*, \bar{\epsilon}_u, \bar{\phi}_{S_u^*}) \le f(S_u, \bar{\epsilon}_u, U_{S_u})/\gamma$. And thus

$$f(S_u^*, \bar{\epsilon}_u, \bar{\phi}_{S_u^*}) - f(S_u, \bar{\epsilon}_u, \bar{\phi}_{S_u})/\gamma \le \frac{1}{\gamma}\left(f(S_u, \bar{\epsilon}_u, U_{S_u}) - f(S_u, \bar{\epsilon}_u, \bar{\phi}_{S_u})\right). \quad (14)$$

Building upon previous discussions, when $\mathcal{E}_{u-1}$ holds, we have

$$\begin{aligned}
&\mathbb{E}\Big[f(S_u, \bar{\epsilon}_u, U_{S_u}) - f(S_u, \bar{\epsilon}_u, \bar{\phi}_{S_u})|\mathcal{E}_{u-1}\Big] \\
\overset{(a)}{\le}&\mathbb{E}\Big[\sum_{k=1}^{K} \bar{\epsilon}_u(k) U_{S_u}(v_k) - \sum_{k=1}^{K} \bar{\epsilon}_u(k)\bar{\phi}_{S_u}(v_k)|\mathcal{E}_{u-1}\Big] \\
=&\mathbb{E}\Big[\sum_{k=1}^{K} \bar{\epsilon}_u(k)(U_{S_u}(v_k) - \bar{\phi}_{S_u}(v_k))|\mathcal{E}_{u-1}\Big] \\
=&\mathbb{E}\Big[\sum_{i=1}^{K} (\bar{\epsilon}_u(i) - \bar{\epsilon}_u(i+1)) \sum_{k=1}^{i} (U_{S_u}(v_k) - \bar{\phi}_{S_u}(v_k))|\mathcal{E}_{u-1}\Big] \\
\overset{(b)}{\le}&p_v \mathbb{E}\Big[\sum_{i=1}^{K}\sum_{k=1}^{i} (U_u(v_k) - L_u(v_k))|\mathcal{E}_{u-1}\Big] \\
\le&2p_v \mathbb{E}\Big[\sum_{i=1}^{K}\sum_{k=1}^{i} \rho_u(v_k)|\mathcal{E}_{u-1}\Big],
\end{aligned}$$

where $\bar{\epsilon}_u(K+1) = 0$, and $p_v = \max_{u=1:n} \max_{k=1:K} \bar{\epsilon}_u(k) - \bar{\epsilon}_u(k+1)$ denotes the maximal difference of termination weights between two consecutive positions for users. The inequality (a) holds by the fact that $\sum_{k=1}^{K} x_k \ge 1 - \Pi_{k=1}^{K}(1-x_k)$ for $x \in [0,1]^K$, and $U_{S_u}(v_k)$ denotes the upper bound of item v_k in the list S_u. Inequality (b) is derived by definition of p_v and $\bar{\phi}_u(v_k) \ge L_u(v_k)$. The last inequality follows from the definitions of L_u and U_u.

Applying Eq. (14) and substitute the above inequality back into Eq. (13) we have

$$\begin{aligned}
&\mathcal{G}_\gamma(n) \\
\le&P(\mathcal{E}_{u-1})\frac{2p_v}{\gamma}\mathbb{E}\Big[\sum_{u=1}^{n}\sum_{i=1}^{K}\sum_{k=1}^{i} \rho_u(v_k)|\mathcal{E}_{u-1}\Big] + \sum_{u=1}^{n} P(\bar{\mathcal{E}}_{u-1}) \\
\le&\frac{2p_v}{\gamma}\mathbb{E}\Big[\sum_{u=1}^{n}\sum_{i=1}^{K}\sum_{k=1}^{i} \rho_u(v_k)\Big] + \sum_{u=1}^{n} P(\bar{\mathcal{E}}_{u-1}).
\end{aligned}$$

The regret bound can be obtained by adapting the worst-case bound on $\sum_{u=1}^{n}\sum_{k=1}^{K} \rho_u(v_k)$ from [41], and a bound on $P(\bar{\mathcal{E}})$. In particular, our $\sum_{u=1}^{n}\sum_{i=1}^{K}\sum_{k=1}^{i} \rho_u(v_k) \le K\sum_{u=1}^{n}\sum_{k=1}^{K} \rho_u(v_k)$.

Lemma 5.2: (Lemma 2 in [41]) The following worst-case bound holds

$$\sum_{u=1}^{n}\sum_{k=1}^{K} \rho_u(v_k) \le K\sqrt{\frac{q_0 n \log(1 + \frac{nK}{q_0\sigma^2})}{\log(1 + \frac{1}{\sigma^2})}}. \quad (15)$$

Lemma 5.3: (Lemma 3 in [41]) For any u, $\sigma > 0$, $\delta \in (0,1)$, and

$$s \ge \frac{1}{\sigma}\sqrt{q_0 \log\left(1 + \frac{nK}{q_0\sigma^2}\right) + 2\log\left(\frac{1}{\delta}\right)} + \|\omega_*\|_2,$$

we have $P(\bar{\mathcal{E}}_u) \le \delta$.

Specifically, let $\delta = 1/n$, we have $P(\bar{\mathcal{E}}_u) \le 1/n$ for all u. Then substituting the worst-case bound back and applying Lemma 3 yields our claimed result.

B. Efficiency Analysis

Then we analyze the efficiency of RAPID. The main computational cost of RAPID consists of two parts: the listwise relevance estimator and the personalized diversity estimator. Let c_0 be the constant of one-step inference time for LSTM. Then the listwise relevance estimator achieves a $\mathcal{O}(c_0 L)$ complexity for each request, where L is the maximum length of the initial ranking list. For the additional personalized diversity estimator, the complexity is $\mathcal{O}(c_0 m D)$, where m is the number of topics, and D is the maximum length of behavioral sequences. Therefore, the overall complexity of RAPID is $\mathcal{O}(c_0(L + mD))$. Our experimental study shows that good performance can be achieved at $D = 5$ or 10, and the number of topics m is usually less than 100. Therefore, the computation cost is inexpensive. Moreover, the computation of m behavioral sequences can be made parallel to further accelerate the process.

VI. Conclusions

In this work, we provide a new perspective to rethink the optimization of the utility for re-ranking. The utility of the list is a comprehensive concept and depends not only on the relevance of items but also on the list diversity. Purely improving relevance or diversity does not necessarily bring the optimal utility, but recommending diverse and fresh items originating from the users' diversity preference shows the potential of increasing the utility. As a consequence, we propose a novel method, RAPID, for re-ranking with personalized diversification. Instead of equally increasing diversity for all users, we propose to provide diverse re-ranking lists that are aware of individual users' diversity interests. We exploit rich user behavior history and model the user-specific preference distribution over all topics. The proposed framework jointly learns the relevance and the diversity, and automatically manages the tradeoff between them. Extensive experiments show that RAPID outperforms state-of-the-art algorithms, achieving the highest utility and the best relevance-diversity tradeoff.

REFERENCES

[1] I. Bello, S. Kulkarni, S. Jain, C. Boutilier, E. Chi, E. Eban, X. Luo, A. Mackey, and O. Meshi, "Seq2slate: Re-ranking and slate optimization with rnns," 2019.

[2] C. Pei, W. Ou, D. Pei, Y. Zhang, Y. Zhang, F. Sun, X. Lin, H. Sun, J. Wu, P. Jiang, and J. Ge, "Personalized re-ranking for recommendation," in *RecSys*, 2019.

[3] Y. Feng, Y. Gong, F. Sun, Q. Liu, and W. Ou, "Revisit recommender system in the permutation prospective," 2021.

[4] W. Liu, Q. Liu, R. Tang, J. Chen, X. He, and P. Heng, "Personalized re-ranking with item relationships for e-commerce," in *CIKM*, 2020.

[5] W. Liu, Y. Xi, J. Qin, F. Sun, B. Chen, W. Zhang, R. Zhang, and R. Tang, "Neural re-ranking in multi-stage recommender systems: A review," in *International Joint Conference on Artificial Intelligence*, 2022.

[6] J. Hron, K. Krauth, M. Jordan, and N. Kilbertus, "On component interactions in two-stage recommender systems," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2744–2757, 2021.

[7] Q. Ai, K. Bi, J. Guo, and W. Croft, "Learning a deep listwise context model for ranking refinement," in *SIGIR*, 2018.

[8] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[9] X. Qin, Z. Dou, and J.-R. Wen, "Diversifying search results using self-attention network," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1265–1274.

[10] M. Wilhelm, A. Ramanathan, A. Bonomo, S. Jain, E. H. Chi, and J. Gillenwater, "Practical diversified recommendations on youtube with determinantal point processes," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 2165–2173.

[11] L. Yan, Z. Qin, R. K. Pasumarthi, X. Wang, and M. Bendersky, "Diversification-aware learning to rank using distributed representation," in *Proceedings of the Web Conference 2021*, 2021, pp. 127–136.

[12] Q. Wu, Y. Liu, C. Miao, B. Zhao, Y. Zhao, and L. Guan, "Pd-gan: Adversarial learning for personalized diversity-promoting recommendation." in *IJCAI*, vol. 19, 2019, pp. 3870–3876.

[13] Z. Jiang, J.-R. Wen, Z. Dou, W. X. Zhao, J.-Y. Nie, and M. Yue, "Learning to diversify search results via subtopic attention," in *SIGIR*, 2017, pp. 545–554.

[14] L. Xia, J. Xu, Y. Lan, J. Guo, and X. Cheng, "Modeling document novelty with neural tensor network for search result diversification," in *SIGIR*, 2016, pp. 395–404.

[15] W. Liu, J. Guo, N. Sonboli, R. Burke, and S. Zhang, "Personalized fairness-aware re-ranking for microlending," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 467–471.

[16] N. Sonboli, F. Eskandanian, R. Burke, W. Liu, and B. Mobasher, "Opportunistic multi-aspect fairness through personalized re-ranking," in *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, 2020, pp. 239–247.

[17] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1059–1068.

[18] J. Qin, W. Zhang, X. Wu, J. Jin, Y. Fang, and Y. Yu, "User behavior retrieval for click-through rate prediction," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 2347–2356.

[19] H. Qian, Q. Wu, K. Zhang, Z. Zhang, L. Gu, X. Zeng, J. Zhou, and J. Gu, "Scope-aware re-ranking with gated attention in feed," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 804–812.

[20] L. Pang, J. Xu, Q. Ai, Y. Lan, X. Cheng, and J.-R. Wen, "Setrank: Learning a permutation-invariant ranking model for information retrieval," in *SIGIR*, 2020.

[21] J. Huang, Y. Li, S. Sun, B. Zhang, and J. Huang, "Personalized flight itinerary ranking at fliggy," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2541–2548.

[22] Z. Lin, S. Zang, R. Wang, Z. Sun, C. Xu, and C.-K. Kwoh, "Attention over self-attention: Intention-aware re-ranking with dynamic transformer encoders for recommendation," *arXiv preprint arXiv:2201.05333*, 2022.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1998, pp. 335–336.

[25] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 22–32.

[26] A. Ashkan, B. Kveton, S. Berkovsky, and Z. Wen, "Optimal greedy diversity for recommendation," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[27] C. Sha, X. Wu, and J. Niu, "A framework for recommending relevant and diverse items." in *IJCAI*, vol. 16, 2016, pp. 3868–3874.

[28] L. Chen, G. Zhang, and H. Zhou, "Fast greedy map inference for determinantal point process to improve recommendation diversity," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 5627–5638.

[29] Y. Huang, W. Wang, L. Zhang, and R. Xu, "Sliding spectrum decomposition for diversified recommendation," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3041–3049.

[30] Y. Shi, X. Zhao, J. Wang, M. Larson, and A. Hanjalic, "Adaptive diversification of recommendation results via latent factor portfolio," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012, pp. 175–184.

[31] W. Wu, L. Chen, and L. He, "Using personality to adjust diversity in recommender systems," in *Proceedings of the 24th ACM conference on hypertext and social media*, 2013, pp. 225–229.

[32] T. Di Noia, V. C. Ostuni, J. Rosati, P. Tomeo, and E. Di Sciascio, "An analysis of users' propensity toward diversity in recommendations," in *Proceedings of the 8th ACM Conference on Recommender Systems*, 2014, pp. 285–288.

[33] A. Yan, C. Dong, Y. Gao, J. Fu, T. Zhao, Y. Sun, and J. McAuley, "Personalized complementary product recommendation," 2022.

[34] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[35] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 273–278.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] G. Hiranandani, H. Singh, P. Gupta, I. A. Burhanuddin, Z. Wen, and B. Kveton, "Cascading linear submodular bandits: Accounting for position bias and diversity in online learning to rank," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 722–732.

[38] C. Li, H. Feng, and M. d. Rijke, "Cascading hybrid bandits: Online learning to rank for relevance and diversity," in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 33–42.

[39] Y. Yue and C. Guestrin, "Linear submodular bandits and their application to diversified retrieval," in *Neural Information Processing Systems*, 2011.

[40] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 661–670.

[41] S. Zong, H. Ni, K. Sung, N. R. Ke, Z. Wen, and B. Kveton, "Cascading bandits for large-scale recommendation problems," in *UAI*, 2016.

[42] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[43] Y. Jia, H. Wang, S. Guo, and H. Wang, "Pairrank: Online pairwise learning to rank by divide-and-conquer," in *Proceedings of the Web Conference 2021*, 2021, pp. 146–157.

[44] X. Dai, J. Hou, Q. Liu, Y. Xi, R. Tang, W. Zhang, X. He, J. Wang, and Y. Yu, "U-rank: Utility-oriented learning to rank with implicit feedback," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2373–2380.

[45] F. Guo, C. Liu, and Y. M. Wang, "Efficient multiple-click models in web search," in *Proceedings of the second acm international conference on web search and data mining*, 2009, pp. 124–131.

[46] W. Liu, S. Li, and S. Zhang, "Contextual dependent click bandit algorithm for web recommendation," in *International Computing and Combinatorics Conference*. Springer, 2018, pp. 39–50.

[47] S. Katariya, B. Kveton, C. Szepesvari, and Z. Wen, "Dcm bandits: Learning to rank with multiple clicks," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1215–1224.
[48] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.
[49] M. Kaminskas and D. Bridge, "Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 7, no. 1, pp. 1–42, 2016.
[50] M. Ge, C. Delgado-Battenfeld, and D. Jannach, "Beyond accuracy: evaluating recommender systems by coverage and serendipity," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 257–260.
[51] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction." in *KDD*, 2018.
[52] T. Joachims, "Training linear svms in linear time," in *KDD*, 2006.
[53] Z. Hu, Y. Wang, Q. Peng, and H. Li, "Unbiased lambdamart: An unbiased pairwise learning-to-rank algorithm," in *WWW*, 2019.
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *The International Conference on Learning Representations*, 2014.
[55] S. Abbar, S. Amer-Yahia, P. Indyk, and S. Mahabadi, "Real-time recommendation of diverse related articles," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 1–12.
[56] J. Rao, A. Jia, Y. Feng, and D. Zhao, "Taxonomy based personalized news recommendation: Novelty and diversity," in *International Conference on Web Information Systems Engineering*. Springer, 2013, pp. 209–218.
[57] S.-T. Yuan and Y. W. Tsao, "A recommendation mechanism for contextualized mobile advertising," *Expert systems with applications*, vol. 24, no. 4, pp. 399–414, 2003.
[58] J. Wang, W. Zhang, S. Yuan *et al.*, "Display advertising with real-time bidding (rtb) and behavioural targeting," *Foundations and Trends® in Information Retrieval*, vol. 11, no. 4-5, pp. 297–435, 2017.
[59] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," *Advances in neural information processing systems*, vol. 24, pp. 2312–2320, 2011.
[60] W. Chu, L. Li, L. Reyzin, and R. Schapire, "Contextual bandits with linear payoff functions," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 208–214.