# Diversity-aware Deep Ranking Network for Recommendation

Zihong Wang*
Beijing University of Posts and Telecommunications
Beijing, China
wzhyt1@bupt.edu.cn

Yingxia Shao†
Beijing University of Posts and Telecommunications
Beijing, China
shaoyx@bupt.edu.cn

Jiyuan He
Meituan Inc.
Beijing, China
hejiyuan@meituan.com

Jinbao Liu
Meituan Inc.
Beijing, China
liujinbao@meituan.com

Shitao Xiao
Beijing University of Posts and Telecommunications
Beijing, China
stxiao@bupt.edu.cn

Tao Feng
Meituan Inc.
Beijing, China
fengtao02@meituan.com

Ming Liu
Meituan Inc.
Beijing, China
liuming04@meituan.com

## ABSTRACT

Diversity is a vital factor in recommendation systems. Improving the diversity in recommendations helps broaden users' horizons, bring good user experience and promote the enterprises' sales. In the past years, many efforts have been devoted to optimizing the diversity in the matching stage and the re-ranking stage of the recommendation system, but few in the ranking stage. The ranking stage is the intermediate stage of the recommendation system. Improving the diversity of the ranking stage can preserve the diversity of the matching stage, and provide a more diversified list for the re-ranking stage. Besides, the ranking models are able to achieve a better balance between accuracy and diversity. In this paper, we aim to improve the diversity in the ranking stage. To address the diversity challenges posed by the pointwise ranking model and biased user interaction history, we propose a Diversity-aware Deep Ranking Network by carefully designing two diversity-aware components that are diversity-aware listwise information fusion and balanced weighting loss. We conduct both offline and online experiments, and the results demonstrate that our proposed model effectively improves the recommendation diversity in the ranking stage while maintaining the accuracy. Moreover, the new model achieves 1.27%, 2.30% and 1.98% improvements in VBR, GMV and Coverage in Meituan, one of the world's largest E-commerce platforms.

*Work was done during Zihong Wang's internship at Meituan.
†Corresponding author.


*CIKM '23, October 21–25, 2023, Birmingham, United Kingdom.*

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614848

## CCS CONCEPTS

• **Information systems** → **Personalization**.

## KEYWORDS

recommendation systems; diversified recommendation; deep ranking network

**ACM Reference Format:**
Zihong Wang, Yingxia Shao, Jiyuan He, Jinbao Liu, Shitao Xiao, Tao Feng, and Ming Liu. 2023. Diversity-aware Deep Ranking Network for Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614848

## 1 INTRODUCTION

Recommendation system is a useful solution to overcome the information overload problem [28]. Nowadays, it has been successfully applied to e-commerce [11, 19], news feeds [33, 34] and other fields. A typical recommendation system contains three stages: matching, ranking, and re-ranking [39]. Each stage is designed to help users find their interested items from the candidates generated by the previous stage and the matching stage selects the items from the entire item set.

Accuracy is the primary goal of recommendation systems and an accurate recommendation system can bring substantial earnings. There are many works focus on improving the accuracy of the recommendation systems [9, 12, 27]. However, accurate recommendation might bring a poor user experience. For example, when we use shopping apps (e.g., Taobao, JD.Com, etc.), we find that the apps repeatedly recommend the items we have browsed or bought before. Users are tired of the repeated contents. Increasing the diversity of recommendation results can help improve user satisfaction [35].

Diversity needs to be considered at every stage of a recommendation system [14]. Recently, many efforts have been made to improve the diversity in the matching stage [3, 21, 39] and re-ranking

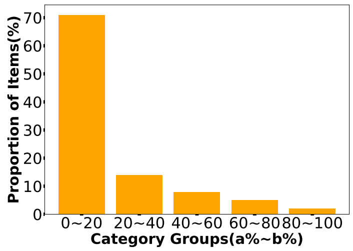



**Figure 1: Distribution of interacted item categories from Meituan dataset**

stage [1, 2, 5], but few in the ranking stage. Improving diversity in the ranking stage has the following benefits:

- The ranking stage is the intermediate stage of the recommendation system, which plays a connection for the whole system. Therefore, improving the diversity of the ranking stage can preserve the diversity of the matching stage [36], and also provide a more diversified list for the re-ranking stage [39].
- The ranking models can improve the recommendation diversity while maintaining the recommendation accuracy. In other words, they achieve a better balance between accuracy and diversity compared with the re-ranking models [18, 39]. And the ranking models also have lower time cost than re-ranking models.

Therefore, it is necessary to maintain a diversified ranking stage. However, existing ranking models [9, 23, 40] are diversity-unaware and are not easy to generate a diversified list. The challenges are two-fold:

(1) Most of them are models [23, 40] with pointwise components only, which makes it difficult to rank out a diversified list. First, pointwise model often has symmetrical network structures and information, which leads it to generate close scores for very similar items [37], so that similar items are difficult to separate, and the diversity of the recommendation list will be reduced. Second, the re-ranking model can re-rank a diverse list because it utilizes listwise information to optimize the diversity distribution of the recommendation list [1, 5]. Therefore, how to make the pointwise ranking model rank out a diversified list is a challenging task.

(2) Most ranking models [22, 23, 40] encode user interaction history to learn user interests, but the interaction history has a long-tail distribution in user-category interactions which destroys the diversity of the ranking stage. In Figure 1, we plot the distribution of the interacted categories. We see that more than 70% of items interacted by users belong to 20% of interacted item categories (i.e., popular item categories), while most categories only have few user interactions. As a result, recommendation models trained on a dataset with such long-tail category distribution would easily overfit on a small fraction of popular categories, and produce a sub-optimal diversity performance for the rest of the large chunk of tail categories. Deploying such models in real-world applications would lead the recommendation list occupied by the popular categories, while the tail categories are rarely exposed, damaging the diversity of the recommendation list.

In this work, we aim to improve the diversity of the ranking stage, and propose a **D**iversity-aware **D**eep **R**anking **N**etwork (DDRN). In DDRN, we design two diversity-aware components: diversity-aware listwise information fusion and balanced weighting loss, to address the aforementioned two challenges, respectively. We conduct extensive offline experiments on real-world datasets and online testing in one of the world's largest E-commerce platforms. The offline results demonstrate that DDRN outperforms state-of-the-art baselines in both accuracy and diversity. The online results showcase that DDRN obtains 1.27%, 2.30% and 1.98% improvements in Visited-Buy Rate (VBR), Gross Merchandise Volume (GMV) and Coverage, which is very significant considering the huge turnover of the E-commerce platform.

In summary, our contributions are three-fold:

- We propose a model to improve both the diversity and accuracy of the ranking stage, which not only preserves the diversity of the matching stage, but also provides a diversified list for the re-ranking stage. So our model is conducive to improving the final diversity of the entire recommendation pipeline.
- Our proposed model DDRN achieves a better balance between accuracy and diversity compared with matching models and re-ranking models. DDRN improves the recommendation diversity in the ranking stage by generating a diverse embedding with listwise information and balancing the training gradients of interaction categories.
- The proposed model outperforms state-of-the-art baselines in offline experiments and achieves significant improvements in VBR, GMV and Coverage in one of the world's largest E-commerce platforms.

## 2 PRELIMINARIES AND RELATED WORK

### 2.1 Recommendation Pipeline

As illustrated in Figure 2, an industrial recommender system typically consists of three stages: (1) matching stage, (2) ranking stage and (3) re-ranking stage. The matching stage usually generates several hundreds of candidates that user might be interested in from a large item pool. Then, usually complicated deep learning models are adopted in ranking stage to estimate interaction probability and the top dozens of items are selected. In the re-ranking stage, the recommendation list generated from the ranking stage is usually re-ranked to satisfy some additional needs.

As for diversity, most of existing diversity solutions are deployed in the matching stage [3, 6, 13, 30] and the re-ranking stage [1, 2, 5], but few in the ranking stage. Diversification in the ranking stage is also of much importance. With ranking models unaware of diversification signals, the diversity of the generated list from the matching stage will decline after going through the ranking stage, and the diversity of the re-ranking stage will also be limited by the diversity-unaware ranking stage [36]. Therefore, our diversity solutions aim at diversification in the ranking stage, preserve the diversity of the upstream matching stage, and provide a more diversified list for the re-ranking stage.

### 2.2 Problem Formulation

In recommendation tasks, let the user set be $U$, the item set be $I$, and the category set of $I$ be $C$. And for our diversification task, we bring the side information(features) of items. Let $R$ be the side information

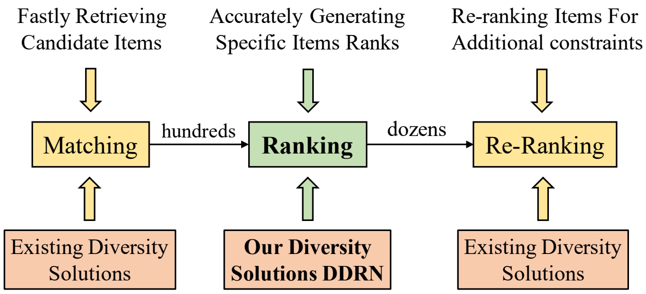



**Figure 2: A typical recommendation pipeline. DDRN focuses on the ranking stage, which aims to improve the diversity of the ranking stage while maintaining accuracy.**

of item set $I$. As mentioned before, industrial recommendation systems usually consist of three stages: matching stage, ranking stage and re-ranking stage. For a user $u$, we denote the item set from the matching stage as the $M = \{m_1, m_2, ..., m_l\}$, which is diversified. The interaction history of a user $u$ used by the ranking stage is denoted by $H = \{h_1, h_2, ..., h_t\}$. Through embedding lookup table, we get embedding vectors $\boldsymbol{E_m}$ of $M$ by concating the id and side information embeddings,

$$\begin{aligned}\boldsymbol{E_m} = \{&[\boldsymbol{e_{m_{1,id}}}; \boldsymbol{e_{m_{1,r_1}}}; ...; \boldsymbol{e_{m_{1,r_k}}}], [\boldsymbol{e_{m_{2,id}}}; \boldsymbol{e_{m_{2,r_1}}}; ...; \boldsymbol{e_{m_{2,r_k}}}],\\ &..., [\boldsymbol{e_{m_{l,id}}}; \boldsymbol{e_{m_{l,r_1}}}; ...; \boldsymbol{e_{m_{l,r_k}}}]\} = \{\boldsymbol{e_{m_1}}, \boldsymbol{e_{m_2}}, ..., \boldsymbol{e_{m_l}}\}\end{aligned} \tag{1}$$

embedding vectors $\boldsymbol{E_h}$ of $H$ by concating the id and side information embeddings.

$$\begin{aligned}\boldsymbol{E_h} = \{&[\boldsymbol{e_{h_{1,id}}}; \boldsymbol{e_{h_{1,r_1}}}; ...; \boldsymbol{e_{h_{1,r_k}}}], [\boldsymbol{e_{h_{2,id}}}; \boldsymbol{e_{h_{2,r_1}}}; ...; \boldsymbol{e_{h_{2,r_k}}}],\\ &..., [\boldsymbol{e_{h_{t,id}}}; \boldsymbol{e_{h_{t,r_1}}}; ...; \boldsymbol{e_{h_{t,r_k}}}]\} = \{\boldsymbol{e_{h_1}}, \boldsymbol{e_{h_2}}, ..., \boldsymbol{e_{h_t}}\}\end{aligned} \tag{2}$$

where $id$ denote the id features of item, and $r_i$ denote the $i_{th}$ side information of item.

In this work, we improve the diversity of recommendation list generated in the ranking stage while maintaining accuracy. The diversity of recommendation list is defined by the number of categories in the list and the intra-list distance (ILD) which is defined in Eq. 15. The more categories and longer ILD in the recommendation list, the higher diversity is.

### 2.3 Related work

In this subsection, we review existing works improving the diversity of recommendation. Most of them focus on the matching stage and re-ranking stage.

In the matching stage, we use embedding representation to model users and items, and rank the items based on their similarity to the user in the embedding space. The key of improving the diversity in the matching stage is to learn a high-quality user embedding model which captures the diverse interests of users. For example, DGCN [39] applies a re-balanced neighbor sampling strategy to the GCN and obtains a user embedding by aggregating more diverse neighbors. Arguing that a single embedding for each user is not expressive enough, some works are proposed to adopt multiple embeddings to represent users' multiple interests [3, 16, 21]. A typical model is ComiRec [3], which uses capsule networks and dynamic routing to model user behaviors into multiple representations of interests, and matches items under each interest. There are also works utilizing generative adversarial network [35], and reinforcement learning [7, 15] to learn a trade-off between accuracy and diversity.

In the re-ranking stage, we divide the models into two groups. One is the models based on post-processing heuristics [1–3, 29], which maximize an object function by considering both the accuracy and diversity. MMR [2] represents accuracy and diversity with independent metrics and uses a hyper-parameter to adjust their balance and maximize the margin relevance. The other group is the models based on the determinantal point process (DPP). DPP [5] constructs a kernel matrix through the relevance matrix and similarity matrix, uses MAP strategy [5] to greedily generate diversified and relevant list. Other works [8, 20, 32] adopt DPP to improve recommendation diversity for different tasks. For example, DivKG [20] integrates knowledge graph and DPP to improve the diversity of recommendation.

Different from the matching stage and re-ranking stage, ranking stage usually adopts a more complex deep neural network to rank items, such as DIN [40], SIM [25], MMOE [23]. Most of them focus on improving recommendation accuracy, and there is little work devoted to the recommendation diversity. Our work aims to enhance diversity in the ranking stage.

## 3 DIVERSITY-AWARE DEEP RANKING NETWORK

We propose a Diversity-aware Deep Ranking Network (DDRN) to generate an accurate and diversified recommendation list in the ranking stage. Figure 3 presents the model architecture of DDRN. For each user and a target item, the user interaction encoding module learns user interests from the user interacted items and guarantees the recommendation accuracy. It adopts popular self-attention mechanism [31] and target attention mechanism [40] to encode the user interaction sequence and get the user preference $\boldsymbol{e_h}$. The diversity-aware listwise information fusion module encodes the matching item set $M$ and generates a listwise personalized embedding $\boldsymbol{e_m}$ for each target item, separating the similar items and optimizing the diversity distribution of the recommendation list. The target item embedding $\boldsymbol{e_c} = [\boldsymbol{e_{c,id}}; \boldsymbol{e_{c,r_1}}; \boldsymbol{e_{c,r_2}} ...; \boldsymbol{e_{c,r_k}}]$ is generated by concating id and side information embedding of item. The user profile features and context features are embedded as $\boldsymbol{e_p}$ and $\boldsymbol{e_t}$ individually. Finally, the predicting layer concatenates all the generated embeddings (i.e., $\boldsymbol{e_h}, \boldsymbol{e_c}$ $\boldsymbol{e_m}, \boldsymbol{e_p}, \boldsymbol{e_t}$) together to obtain the overall representation vector and adopts a MLP to generate the prediction $y'$ of the target item. During the training of DDRN, we optimize the network with a balanced weighting loss to eliminate the bias of training gradients between item categories with different interacted frequency. Through the combination of diversity-aware listwise information fusion and balanced weighting loss, we optimize the ability of the model to distinguish similar items, eliminate data distribution bias, and improve the accuracy and diversity of recommendations.

### 3.1 Diversity-aware Listwise Information Fusion

As mentioned above, the ranking models are often diversity-unaware and only have pointwise components, so they generate close scores

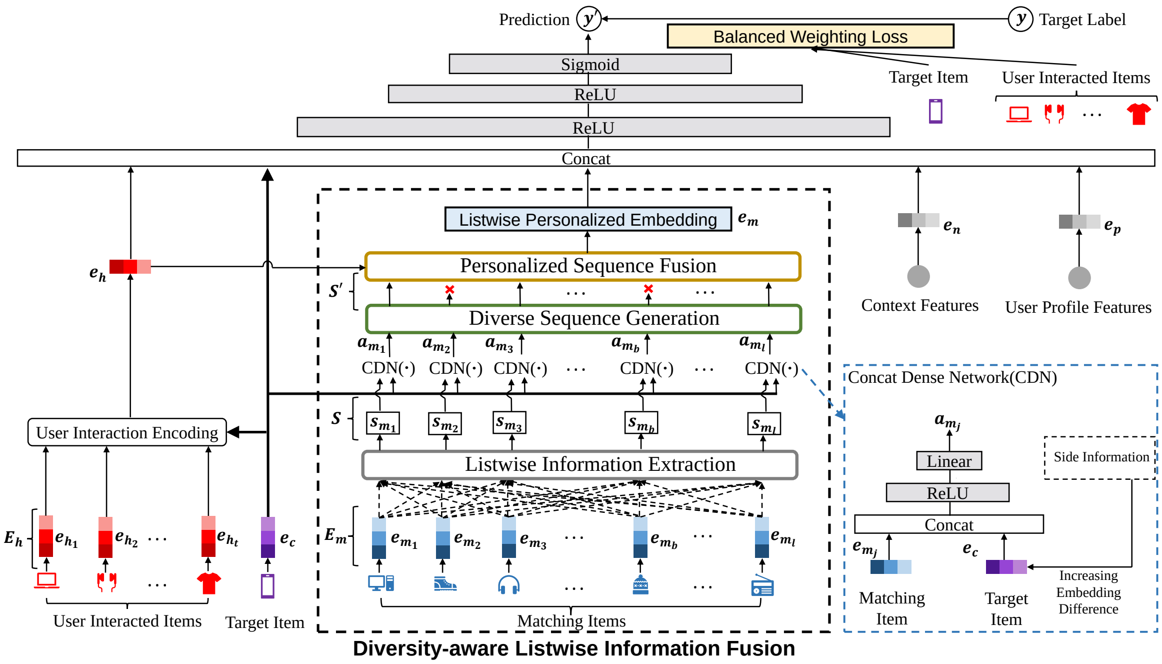



**Figure 3: Model architecture of DDRN.**

for very similar items and it is difficult to optimize the diversity distribution of the recommendation list without listwise information. Inspired by the re-ranking models' [1, 2, 5] use of listwise information to optimize diversity, we design a listwise-aware module "Diversity-aware Listwise Information Fusion" to solve the challenge. First, we introduce the information of the matching item set $\boldsymbol{M}$ and encode the $\boldsymbol{M}$ into $\boldsymbol{S}$ through the self-attention mechanism to extract the listwise information of the recommendation list. Then, in order to generate more different feature representations for similar items, we filter the embedding sequence $\boldsymbol{S}$ to the $\boldsymbol{S'}$ by selecting most influential embeddings to the target item to enhance the difference of the embedding sequence between similar items. Finally, to ensure the accuracy of the generated embedding, we use user preference embedding generated in the user interaction encoding module to aggregate the embedding sequence $\boldsymbol{S'}$ and generate the final listwise personalized embedding.

*3.1.1 Listwise Information Extraction.* Matching item set $\boldsymbol{M}$ from the matching stage is often generated through the hot list, user interests, collaborative filtering and other channels. This set is highly diverse and covers items that user might be interested in. Besides, the matching item set contains the listwise information of the entire candidate item list, which is helpful for the diversification task [37]. Therefore, the goal of listwise information extraction module is to extract the listwise information of each item in the list. To achieve this goal, we adopt the self-attention mechanism introduced in Transformers [31]. It directly models the mutual influences for any items in the total list. And the Transformer can capture more interactions between items that are far away from each other in the matching item set, which is helpful for diversity optimization.

Specifically, for a user $u$, given the embeddings of the matching item set $\boldsymbol{E_m} = \{\boldsymbol{e_{m_1}}, \boldsymbol{e_{m_2}}, ..., \boldsymbol{e_{m_l}}\}$, each embedding of $\boldsymbol{E_m}$ is formed by concating the embeddings of id and side information of the item. And we use a self-attention encoder(SelfAttnEnc) to process the embedding list $\boldsymbol{E_m}$ and get the output embedding sequence $S = \{\boldsymbol{s_{m_1}}, \boldsymbol{s_{m_2}}, ..., \boldsymbol{s_{m_l}}\}$ as follows:

$$
\begin{aligned}
S &= SelfAttnEnc(\boldsymbol{E_m}) = softmax(\frac{\boldsymbol{QK}^T}{\sqrt{d}})\boldsymbol{V}, \\
\boldsymbol{Q} &= \boldsymbol{E_m W^Q}, \boldsymbol{K} = \boldsymbol{E_m W^K}, \boldsymbol{V} = \boldsymbol{E_m W^V},
\end{aligned}
\tag{3}
$$

where $\boldsymbol{W^Q}, \boldsymbol{W^K}, \boldsymbol{W^V}$ are learnable parameters, $d$ is the dimension of $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$, $softmax$ is the activation function.

After the $\boldsymbol{M}$ is encoded into $\boldsymbol{S}$ by the self-attention mechanism, the diversity information and listwise information of the entire candidate list are encoded into $S$. And the $\boldsymbol{S}$ is helpful for enriching the final overall representation and optimizing the diversity distribution of the recommendation list.

*3.1.2 Diverse Sequence Generation.* Similar items get close scores due to similar feature representations in the ranking model [37]. Therefore, the goal of the diverse sequence generation module is to enhance the difference of feature representations between similar items. Since the embedding sequence $\boldsymbol{S}$ is the same for all target items, there is little difference among output representations of all target items if $S$ is directly encoded. We need to generate different embedding sequence for each target item. Each item in the embedding sequence $\boldsymbol{S}$ has a different influence on the prediction of the target item [40]. Therefore, we filter the embedding sequence $\boldsymbol{S}$ by selecting top-$K$ items which are most influential to the target item prediction from the $S$. And different target item has different influential items, making the filtered embedding sequence different for each target item.

To achieve this goal, we first use concat dense network [17] to compute the contribution of items to the prediction of target item, and select $K$ items with the most significant contributions to form the filtered sequence. Concretely, given $\boldsymbol{S}$ and the embedding of the target item $\boldsymbol{e_c}$ by concating embeddings of id and side information of target item, we calculate the influential score $\boldsymbol{s_{m_j}}$ of item $j$ in $\boldsymbol{S}$ to the target item as follows:

$$\boldsymbol{a_{m_j}} = CDN(\boldsymbol{e_c}, \boldsymbol{s_{m_j}}) = ReLU(\boldsymbol{W^C}[\boldsymbol{e_c}; \boldsymbol{s_{m_j}}]), \tag{4}$$

where $CDN$ is the concat dense network, $ReLU$ is an activation function, $\boldsymbol{W^C}$ is a learnable parameter. Then $K$ items with the highest scores are selected to enhance the prediction of the target item and form the filtered embedding sequence. To make the embedding sequences filtered out by similar items more different, we use side information(attribute features) and the id of the item to get the joint embedding of item. The selected items are denoted by $\boldsymbol{S'} = \{\boldsymbol{s_{m_1}}, \boldsymbol{s_{m_2}}, ..., \boldsymbol{s_{m_K}}\}$.

$$\boldsymbol{S'} = \{\boldsymbol{s_{m_j}}, \boldsymbol{a_{m_j}} \in top-K(\boldsymbol{a_{m_j}})\}, \tag{5}$$

*3.1.3 Personalized Sequence Fusion.* As for the diversity, after the listwise information extraction module and the diverse sequence generation module, a diverse sequence $\boldsymbol{S'}$ with listwise information is generated from the matching item set $\boldsymbol{M}$. To ensure the accuracy of the output embedding, we enhance the personalized information of the output embedding. Considering that not all items in the sequence $\boldsymbol{S'}$ are related to user preference, we aggregate the sequence $\boldsymbol{S'}$ through the user preference embedding $\boldsymbol{e_h}$.

To achieve this goal, we adopt the target attention mechanism from DIN [40] for weighted aggregation of sequence $\boldsymbol{S'}$. The target embedding is the user preference embedding $\boldsymbol{e_h}$ generated in the user interaction encoding module. Specifically, we first use the softmax dot product to calculate the related weight of each item to the user preference, then we aggregate $\boldsymbol{S'}$ with weighted sum pooling and get the final output embedding $\boldsymbol{e_m}$.

$$\begin{aligned} \boldsymbol{e_m} &= f(\boldsymbol{e_h}, \boldsymbol{S'}) = \sum_{j=1}^{K} TA(\boldsymbol{e_h}, \boldsymbol{s_{m_j}})\boldsymbol{s_{m_j}}, \\ TA(\boldsymbol{e_h}, \boldsymbol{s_{m_j}}) &= \frac{exp(\boldsymbol{e_h^T}\boldsymbol{s_{m_j}})}{\sum_{i=1}^{k} exp(\boldsymbol{e_h^T}\boldsymbol{s_{m_i}})}. \end{aligned} \tag{6}$$

where $TA$ is the target attention encoder, $\boldsymbol{s_{m_j}}$ is the embedding $j$ in $\boldsymbol{S'}$.

In summary, through the above three modules, we generate a listwise personalized embedding $\boldsymbol{e_m}$. The $\boldsymbol{e_m}$ not only contains the diversity information in the matching item set $\boldsymbol{M}$, but also contains listwise information of the entire candidate item list. And with the diverse sequence generation module, the difference of $\boldsymbol{e_m}$ between similar items will be increased and help generate different scores to separate the similar items. With user preference information enhanced, the $\boldsymbol{e_m}$ will be more relevant to the user preference, ensuring the accuracy of the embedding. Therefore, the listwise personalized embedding $\boldsymbol{e_m}$ is diversified and personalized which can improve both the diversity and accuracy of the ranking stage.

## 3.2 Balanced Weighting Loss

The non-uniform distribution of user interaction categories (e.g., Figure 1) brings bias to the training samples among categories. The sample bias brings the gradient bias in the training process, and the popular categories will dominate the gradient of the training process, so that the recommendation list is occupied by the items from the popular categories, resulting in low diversity (or imbalance) of categories in the list.

In DDRN, we propose a balanced weighting loss to fix the bias of user interaction category. And we adopt different weighting strategies for positive and negative samples.

**For positive samples**, the basic idea of the new loss is that the category with more interaction has a lower weight for its loss while the category with less interaction has a higher weight. We generate this adaptive weight based on the interaction amount of the category.

Following the basic idea, we change the monotonicity of the interaction amount of each category, and make the larger interaction amount of a category less significant. To be concrete, let the category set interacted by a user $u$ be $C_u^+ = \{c_1, c_2, ..., c_z\}$, and the item set interacted by $u$ be $I_u^+$. For an item $i$ interacted by $u$, its category is $c_i$ and user $u$'s interaction amount of category $c_i$ is $|I_{u,c_i}^+|$. The maximal interaction amount of the category among $C_u^+$ is

$$cm = max\{|I_{u,c_j}^+|, c_j \in C_u^+\}. \tag{7}$$

Then we reverse user's interaction amount of sample $i$'s category $c_i$ as below:

$$R|I_{u,c_i}^+| = cm/|I_{u,c_i}^+|. \tag{8}$$

After the above transformation, larger $|I_{u,c_i}^+|$ leads to smaller $R|I_{u,c_i}^+|$. However, directly using $R|I_{u,c_i}^+|$ as the loss weight has some problems. Large $R|I_{u,c_i}^+|$ (i.e., small $|I_{u,c_i}^+|$) weights the category excessively and damages the accuracy of recommendation. Meanwhile, excessively large weight is not conducive to the convergence of the model. Therefore, we scale the $R|I_{u,c_i}^+|$ to a proper interval of $[0, \gamma]$. We first normalize $R|I_{u,c_i}^+|$ with min-max normalization [24] to $\lambda$ as follows:

$$\lambda = \frac{R|I_{u,c_i}^+| - tcn}{tcm - tcn}, \tag{9}$$

where $tcm$ and $tcn$ are the max and min value of reverse interaction amount among $C_u^+$, as defined below

$$tcm = max\{R|I_{u,c_j}^+|, c_j \in C_u^+\}, \tag{10}$$

$$tcn = min\{R|I_{u,c_j}^+|, c_j \in C_u^+\}. \tag{11}$$

And we multiply $\lambda$ and the hyper-parameter $\gamma$ to obtain the final loss weight $\omega$:

$$\omega = \lambda\gamma. \tag{12}$$

**For negative samples**, the loss weighting strategy is opposite to the positive sample, the negative samples from category with more interaction has a higher weight for its loss while the negative samples from category with less interaction has a lower weight. The calculation process of negative sample weight is similar to that of positive sample. As the loss weight and interaction amount of category are positively correlated, the $\lambda$ of negative sample is directly generated by min-max normalization. For a negative sample $x$ of user $u$, its category is $c_x$ and user's interaction amount of category $c_x$ is $|I_{u,c_x}^+|$. The $\lambda$ is generated as follows:

$$\lambda = \frac{|I_{u,c_x}^+| - cn}{cm - cn}, \tag{13}$$

where $cm$ and $cn$ are user's max and min interaction amount of category among $C_u^+$. And the loss weight $\omega$ is also generated by multiplying $\lambda$ and $\gamma$.

**For the training process**, given a training sample $(u, i, y)$, where $y$ represents the interaction ground-truth of user $u$ on item $i$, the diversity-aware weighting loss is defined

$$loss = -\frac{1}{N} \sum_{(i,y)\in\tau} \boldsymbol{\omega} \cdot (y \cdot log(y') + (1-y) \cdot log(1-y')), \quad (14)$$

where $y'$ is the output of DDRN for predicting item $i$, $\tau$ is the training set of size N. Note that the balanced weighting loss depends on the user's interaction history. Therefore, to ensure the effectiveness of weighting, for users with a user interaction history length less than 5, we set the loss weight of all samples to 1, and for samples of users with rich behavior, we use our designed $\omega \in [0, \gamma]$ as loss weight. Consequently, the gradients in the training process among interaction categories are balanced. Similar but negative samples are further separated from positive samples in categories with more user interaction and categories with less user interaction receive more exposure. Besides, with the combination of balanced weighting loss and listwise personalized embedding in Section 3.1, the model leverages the listwise information in embedding to optimize the diversity distribution of the list and the items of each category will be recommended to users more evenly improving the diversity of recommendations.

# 4 EXPERIMENTS

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

- **RQ1:** What is the performance of DDRN compared with state-of-the-art baselines in terms of accuracy and diversity?
- **RQ2:** What is the performance of DDRN in the overall pipeline diversity of the recommendation system?
- **RQ3:** What is the effect of each proposed diversity strategy in DDRN?
- **RQ4:** What is the performance of DDRN to improve diversity by balancing the popular categories and others?
- **RQ5:** What is the performance of the DDRN for the online platform?
- **RQ6:** How is the inference efficiency of the DDRN compared with re-ranking models?
- **RQ7:** What is the impact of each hyper-parameter on the DDRN?

## 4.1 Experimental Setting

*4.1.1 Datasets.* We test our model on two public datasets Taobao[1], MovieLens[2] and one private industrial dataset from Meituan. For the two public datasets, we adopt 10-core version settings [39] in our experiments, which means each item or user has at least 10 interactions. The statistics of three datasets are shown in Table 1. Taobao is an e-commerce dataset with diverse items and user interactions. MovieLens-1M is used in our experiments. Considering MovieLens is a dataset contains rich users' ratings on movies, following the convection [4, 18], we label the movies with ratings between [4, 5] as the interacted movies, and others are the non-interacted movies. The Meituan dataset is constructed from Meituan, one of the world's largest E-commerce platforms. Meituan dataset contains richer features and user behaviors than public datasets, and we take the user behaviors of one week and the related user features forming the dataset.

**Table 1: Statistics of the datasets**

| Datasets | Users | Items | Interactions | Cate. |
|---|---|---|---|---|
| Taobao | 6,134 | 21,223 | 508,976 | 810 |
| MovieLens | 6,040 | 3,367 | 991,365 | 18 |
| Meituan | 3,257,916 | 63,548 | 10,440,315 | 168 |

*4.1.2 Evaluation Metrics.* To evaluate the performance, following metrics are used in our experiments, including accuracy metrics and diversity metrics.

*Accuracy.* We focus on the CTR prediction in the ranking stage, so we take widely used AUC [10, 41], NDCG [4, 12] as the accuracy metrics.

*Diversity.* In this work, we evaluate the diversity with Coverage [39] and ILD [26, 38]. Coverage is the number of categories in the recommendation list. ILD evaluates the diversity by averaging the similarity between every two items in the recommendation list.

$$ILD = \frac{2}{N^2 - 1} \sum_{i,j\in L} d_{ij}, \quad (15)$$

where $N$ is the length of the recommendation list $L$, and $d_{ij}$ is the similarity between item $i$ and $j$.

*4.1.3 Baselines.* To validate the effectiveness of DDRN, we compare our model with following baselines:

- **ComiRec [3]:** ComiRec is a matching model that models user interaction history into multiple user interest embeddings to match items for user.
- **DRN [40]:** DRN is a ranking model that removes all two diversity-aware components from our DDRN(The structure is basically similar to the ranking model DIN).
- **SIM [25]**: SIM is a ranking model that personalizes user interactions by modeling the user long-term behavior sequence.
- **MMOE [23]**: MMOE is a ranking model for multi tasks such as CTR prediction and CVR prediction.
- **MMR [2]:** MMR is a classic re-ranking model for diversified recommendation by maximizing the marginal relevance.
- **DPP [5]:** DPP is a re-ranking model optimizes the trade-off between accuracy and diversity, and uses MAP inference to generate a diversified recommendation list.
- **CATE:** CATE is a re-ranking model that proposed in the ComiRec [3]. CATE balances the accuracy and categories distribution to greedy generate a diversified and accurate recommendation list.
- **PMF [1]:** PMF is a re-ranking model which optimizes the recommendation diversity by balancing three aspects of the recommendation list: the relevance of the items, the coverage of the user's interest and the diversity between them.

ComiRec is a model used in the matching stage; DRN, SIM and MMOE are models used in the ranking stage; MMR, PMF, DPP, and CATE are the ones used in the re-ranking stage.

[1] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
[2] https://grouplens.org/datasets/movielens/

*4.1.4 Settings.* For each dataset, we split all users into training/validation/test sets by the proportion of 8:1:1. In our experiment, we construct a recommendation pipeline with three stages: matching stage, ranking stage and re-ranking stage. For the matching stage, we use ComiRec as the matching model, and each interest of a user independently retrieves top-$N$ candidate items. Thus, ComiRec retrieves a total of $L \cdot N$ items for each user to form the matching item set $M$, where $L$ is the number of user interests. And the matching item set is ranked based on the similarity between item embedding and interest embedding for evaluation. DRN, SIM, MMOE and DDRN are deployed in the ranking stage, they rank all items in the matching item set generated by the ComiRec in the matching stage. MMR, CATE, DPP and PMF are deployed in the re-ranking stage, they re-rank the recommendation list generated by the DRN in the ranking stage.

As for parameters, for all matching and ranking models, we set the embedding dimension =16, the batch size = 256, the initial learning rate = 0.001, the length of user interaction history = 25, and we use Adam as the optimizer to optimize the network. For the matching model ComiRec, we set the number of interests as 4, and each interest retrieves 50 items to form the matching item set. For our DDRN, we set the $K$ which is the number of items selected from the embedding sequence $S$ as 25 for MovieLens and Meituan datasets, 100 for Taobao dataset. And the maximal weight in the balanced weighting loss $\gamma$ is 4 for MovieLens and Meituan datasets, 3 for Taobao dataset. For re-ranking models MMR, CATE and PMF, we set their trade-off parameters as 0.8, 0.01 and 0.5 respectively. For side information, we use category of item for all three datasets. Detailed settings and code can be found in our Github link: https://github.com/azdyt1/DDRN.

## 4.2 Performance Comparison (RQ1)

The overall performance comparison between DDRN and the baselines on three datasets is shown in Table 2. We divide the experimental results into three groups for comparison: DDRN vs ranking models, DDRN vs re-ranking models and DDRN vs matching models. We have the following analysis:

**DDRN vs Ranking models.** Compared DDRN with DRN, SIM and MMOE, we figure out that our DDRN not only significantly improves the diversity of the ranking stage, but also improves the accuracy on three datasets. Taking DRN on Meituan dataset as an example, C@50 and ILD@50 are significantly improved by 16.94% and 6.48%; AUC and NDCG are increased by 1.38% and 2.66%, respectively. The results also demonstrate that DDRN can improve both the diversity and accuracy of recommendation in the ranking stage.

**DDRN vs re-ranking models.** Compared DDRN with DRN+re-ranking models, DDRN performs better than all the re-ranking models on accuracy. DDRN also outperforms the re-ranking models in diversity metrics. Therefore, DDRN can achieve a better balance between accuracy and diversity compared to re-ranking models. Taking DRN+DPP as an example, although DPP can effectively improve the diversity of DRN, it significantly reduces the accuracy of DRN, while DDRN outperforms DRN in terms of accuracy and its diversity is significantly better than DRN at the same time. In addition to the accuracy and diversity performance, DDRN is also superior to the DRN+re-ranking models in terms of inference efficiency. The detailed analysis are described in Section 4.7.

**DDRN vs matching models.** Compared DDRN and matching model ComiRec, most diversity metrics of DDRN on three dataset are better than those of ComiRec, indicating that DDRN can effectively maintain or even enhance the diversity of upstream matching stage. Then, the accuracy metrics of DDRN are significantly better than ComiRec on all three datasets, while ComiRec has the worst accuracy metrics on all three datasets. This is because ComiRec captures a set of items with diverse user-interested categories, but the simple approach of the embedding similarity computation destorys the model accuracy. It requires a smart ranking model like DDRN to maintain both the diversity and accuracy of the recommendations.

In summary, the DDRN siginificantly improves the diversity in the ranking stage while maintaining accuracy compared with all baselines.

**Table 2: The overall performance comparison between DDRN and the baselines on three datasets. For each metric, the best performance is bold, and the second best is marked with an underline. The $'*'$ indicates statistically significant improvement with p-value $<$ 0.05.**

| Datasets | Models | Accuracy | | Diversity | | | |
|---|---|---|---|---|---|---|---|
| | | AUC | NDCG | C@20 | C@50 | ILD@20 | ILD@50 |
| Taobao | ComiRec | 0.6958 | 0.4238 | $\underline{6.8556}$ | **15.0364** | $\underline{0.6832}$ | **0.7710** |
| | DRN | 0.7749 | 0.4640 | 5.3399 | 11.1442 | 0.5834 | 0.6656 |
| | SIM | $\underline{0.7771}$ | 0.4799 | 5.9743 | 12.2707 | 0.6283 | 0.7035 |
| | MMOE | 0.7756 | $\underline{0.4856}$ | 5.9051 | 12.1007 | 0.6306 | 0.6961 |
| | DRN+MMR | 0.7721 | 0.4545 | 5.4861 | 11.5276 | 0.5870 | 0.6668 |
| | DRN+DPP | 0.7313 | 0.4292 | 6.6284 | 13.3201 | 0.6364 | 0.7120 |
| | DRN+CATE | 0.7702 | 0.4610 | 5.9841 | 13.9604 | 0.6480 | 0.7466 |
| | DRN+PMF | 0.7557 | 0.4345 | 6.2984 | 12.7173 | 0.6391 | 0.7013 |
| | DDRN | **0.7784**$^*$ | **0.4899**$^*$ | **6.8873**$^*$ | $\underline{14.5632}^*$ | **0.6898**$^*$ | $\underline{0.7516}^*$ |
| MovieLens | ComiRec | 0.5825 | 0.3541 | 4.8835 | 6.8965 | 0.5870 | 0.5945 |
| | DRN | 0.6056 | 0.3638 | 4.7879 | 7.0051 | 0.6178 | 0.6431 |
| | SIM | 0.6080 | 0.3645 | 4.9620 | 7.2078 | 0.6192 | 0.6529 |
| | MMOE | $\underline{0.6091}$ | $\underline{0.3652}$ | 4.4862 | 6.7379 | 0.5773 | 0.6488 |
| | DRN+MMR | 0.6034 | 0.3627 | 5.0965 | 7.1827 | 0.6316 | 0.6477 |
| | DRN+DPP | 0.5969 | 0.3588 | $\underline{5.6048}$ | 7.3982 | $\underline{0.7094}$ | 0.6572 |
| | DRN+CATE | 0.6033 | 0.3630 | 5.5534 | $\underline{8.2603}$ | 0.7023 | $\underline{0.7180}$ |
| | DRN+PMF | 0.6017 | 0.3570 | 5.0189 | 7.3086 | 0.6418 | 0.6596 |
| | DDRN | **0.6116**$^*$ | **0.3665**$^*$ | **6.3672**$^*$ | **8.5931**$^*$ | **0.7181**$^*$ | **0.7205**$^*$ |
| Meituan | ComiRec | 0.6598 | 0.3049 | 13.5283 | $\underline{26.2520}$ | 0.6220 | $\underline{0.6602}$ |
| | DRN | 0.7235 | 0.3682 | 10.2486 | 22.4839 | 0.5921 | 0.6218 |
| | SIM | 0.7250 | 0.3695 | 10.8781 | 23.3012 | 0.6036 | 0.6280 |
| | MMOE | $\underline{0.7280}$ | $\underline{0.3721}$ | 10.1365 | 22.9256 | 0.6001 | 0.6210 |
| | DRN+MMR | 0.7172 | 0.3629 | 11.2967 | 24.2486 | 0.6082 | 0.6494 |
| | DRN+DPP | 0.7089 | 0.3508 | 11.9939 | 25.1382 | $\underline{0.6234}$ | 0.6512 |
| | DRN+CATE | 0.7130 | 0.3602 | $\underline{13.6831}$ | 24.9796 | 0.6019 | 0.6436 |
| | DRN+PMF | 0.6832 | 0.3398 | 12.9823 | 25.8212 | 0.6184 | 0.6561 |
| | DDRN | **0.7335**$^*$ | **0.3780**$^*$ | **13.7263**$^*$ | **26.2934**$^*$ | **0.6236**$^*$ | **0.6621**$^*$ |

## 4.3 Performance of DDRN in the Recommendation Pipeline (RQ2)

In order to test the performance of DDRN in the whole pipeline of the recommendation system, we conduct experiments on DDRN+re-ranking models. The results are shown in Table 3.

Referring to Table 2 and Table 3, we have the following results. First, more than half of diversity metrics of DDRN on three datasets are slightly better that of ComiRec, which demonstrates that DDRN can retain the diversity of matching stage. Second, compared DRN+re-ranking models with DDRN+re-ranking models, DDRN with an arbitrary re-ranking model is superior to DRN with the corresponding re-ranking model in terms of accuracy and diversity metrics, which demonstrates that DDRN can provide a more relevant and diversified recommendation list for the re-ranking stage. Third, the improvement of the diversity of DDRN by the re-ranking models is significantly less than that of DRN, which shows that DDRN ranks the recommendation list to an ideal diversity state. Besides, compared the DDRN+re-ranking models with ComiRec, the DDRN with an arbitrary re-ranking model achieves better performance than ComiRec on both diversity and accuracy metrics on three datasets, which demonstrates the superiority of DDRN.

To conclude, DDRN can preserve the diversity of matching stage and provide a more relevant and diversified recommendation list for the re-ranking stage.

**Table 3: The overall performance of DDRN+re-ranking modelss on three datasets. For each metric, the best performance is bold, and the second best is marked with an underline.**

| Datasets | Models | Accuracy | | Diversity | | | |
|---|---|---|---|---|---|---|---|
| | | AUC | NDCG | C@20 | C@50 | ILD@20 | ILD@50 |
| Taobao | DDRN+MMR | **0.7756** | 0.4725 | 7.1185 | 15.0474 | 0.6958 | 0.7522 |
| | DDRN+DPP | 0.7378 | 0.4480 | **8.3280** | 16.2984 | **0.7428** | 0.7791 |
| | DDRN+CATE | 0.7726 | **0.4822** | 7.4664 | **16.9249** | 0.7323 | **0.7930** |
| | DDRN+PMF | 0.7608 | 0.4482 | 7.5869 | 15.4901 | 0.7185 | 0.7713 |
| MovieLens | DDRN+MMR | **0.6108** | **0.3641** | 6.9586 | 8.8034 | 0.7495 | 0.7250 |
| | DDRN+DPP | 0.5999 | 0.3608 | 7.1413 | 8.7327 | 0.7578 | 0.7246 |
| | DDRN+CATE | 0.6085 | 0.3649 | **7.1672** | **9.3775** | **0.7978** | **0.7855** |
| | DDRN+PMF | 0.6101 | 0.3649 | 6.4189 | 8.6103 | 0.7356 | 0.7327 |
| Meituan | DDRN+MMR | 0.7198 | 0.3643 | 13.3248 | 26.6093 | 0.6278 | 0.6637 |
| | DDRN+DPP | 0.7149 | 0.3618 | 13.6524 | 26.8736 | 0.6302 | 0.6684 |
| | DDRN+CATE | **0.7240** | **0.3689** | 13.1234 | 26.4376 | 0.6250 | 0.6602 |
| | DDRN+PMF | 0.7093 | 0.3524 | **13.8579** | **27.0764** | **0.6347** | **0.6708** |

## 4.4 Ablation Studies (RQ3)

In order to verify the effectiveness of diversity-aware strategies (i.e., Diversity-aware Listwise Information Fusion (DLIF), Balanced Weighting Loss (BWL), we conduct ablation studies of our DDRN. The results are reported in Table 4.

We see that after removing each diversity-aware strategy, the accuracy and diversity performance of DDRN is decreased, which verifies that the proposed diversity-aware techniques are important to the final performance of DDRN. Among two variants of DDRN, DDRN$_{w/o\ DLIF}$ loses accuracy most on three datasets, and it demonstrates that DLIF can effectively improve the accuracy of the model while improving diversity; The decrease in diversity metrics of DDRN$_{w/o\ DLIF}$ and DDRN$_{w/o\ BWL}$ is very close, indicating that both DLIF and BWL strategies can effectively improve diversity. In addition, the performance of DDRN$_{w/o\ BWL}$ on the MovieLens dataset decreases most among three datasets, and the results imply that BWL has better effect in distinguishing similar items in the data set with few categories (e.g., 18 categories in MovieLens).

**Table 4: Results of Ablation Studies.**

| Datasets | Model | AUC | NDCG | C@50 | ILD@50 |
|---|---|---|---|---|---|
| Taobao | DDRN$_{w/o\ DLIF}$ | 0.7714 | 0.4784 | 13.8833 | 0.7369 |
| | DDRN$_{w/o\ BWL}$ | 0.7771 | 0.4856 | 13.4618 | 0.7149 |
| | DDRN | **0.7784** | **0.4899** | **14.5632** | **0.7516** |
| MovieLens | DDRN$_{w/o\ DLIF}$ | 0.6002 | 0.3598 | 7.4328 | 0.6679 |
| | DDRN$_{w/o\ BWL}$ | 0.6111 | 0.3658 | 7.2023 | 0.6558 |
| | DDRN | **0.6116** | **0.3665** | **8.5931** | **0.7205** |
| Meituan | DDRN$_{w/o\ DLIF}$ | 0.7215 | 0.3670 | 25.1790 | 0.6519 |
| | DDRN$_{w/o\ BWL}$ | 0.7303 | 0.3762 | 25.0077 | 0.6498 |
| | DDRN | **0.7335** | **0.3780** | **26.2934** | **0.6621** |

## 4.5 Case Study (RQ4)

In order to explain DDRN improving the diversity of ranking stage by balancing the popular categories and others, we conduct a case study with DRN and DDRN$_{w/o\ DLIF}$ on MovieLens dataset.

First, we use DRN and DDRN to generate recommendation lists for all users, and take out the top20 items in the recommendation lists of all users. Then we sort the categories by the number of items in them, and divide the categories into five groups at 20% intervals. The results are visualized in Figure 4, where x-axis is the category groups, and y-axis is the proportion of the items in each category group to the total items. We see that in DDRN, with the help of balanced weighting loss, the proportion of items in popular categories decreases, especially the proportion of items in top 20% categories decreased from 69% to 44%. The proportion of items in other categories increases, especially the proportion of items in last 20% categories increased from 2% to 10%. The gap between the top and the last category groups is reduced from 66% to 33%, and the category distribution is more balanced.

Second, for all users, we calculate the position distance between all pairs of items from the same category in the top 20 of recommendation list according to Eq. 16.

$$CD = \frac{\sum_{i \in L} \sum_{j \in L, c_j \in C_i} |p_i - p_j|}{\sum_{i \in L} |C_i|(|C_i| - 1)} \tag{16}$$

where $p_i$ is the position of the item $i$ in the recommendation list $L$. The results of DRN is 5.3927 and that of DDRN is 6.3279. It verifies that with our balanced weighting loss, the distance between items under the same category is increased, indicating that DDRN can effectively separate similar items in the same category.

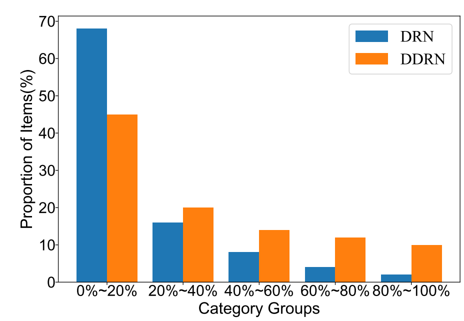



**Figure 4: Comparison of category distribution**

## 4.6 Online Platform Performance (RQ5)

We have deployed our DDRN in Meituan, one of the world's largest E-commerce platforms provided by our industrial partner and conduct the online A/B testing. The control group uses DRN for recommendation and the experimental group uses DDRN.

Table 5 showcases the improvements of DDRN in online A/B test for a week compared with DRN. Our DDRN outperforms DRN on all metrics in 7 days and obtains the average improvement of a week for 1.27% in Visited-Buy Rate (VBR), 2.30% in Gross Merchandise Volume (GMV) and 1.98% in Coverage. This improvement is very significant considering the huge turnover of the E-commerce platform. All the results are statistical significance with p-value $< 0.05$.

These results clearly demonstrate that our proposed model can effectively improve diversity of the ranking model while maintaining the accuracy in the real-world recommendation scenario.

**Table 5: Performance improvements of DDRN in online A/B test compared with DRN.**

| Model | VBR | GMV | Coverage |
|---|---|---|---|
| DDRN | 1.27% | 2.30% | 1.98% |

## 4.7 Inference Efficiency Analysis (RQ6)

As analyzed in Section 4.2, both accuracy and diversity metrics of DDRN are better than those of DRN+re-ranking models. To evaluate the inference efficiency of DDRN and DRN+re-ranking models, we conduct experiments on Taobao dataset. For fairness, we compare the total time of DRN prediction and re-ranking top-$k$ items with that of DDRN prediction and ranking all items with DDRN's prediction. The length of recommendation list is set to 200, and $k$ in re-ranking models are set to 100, 150 and 200 respectively. TP999 and TP50, respectively, refer to the time cost ranks 99.9% and 50% in all time cost, and MEAN is the average of all time cost.

In Table 6, the results showcase that the total time of DDRN is less than that of DRN prediction with re-ranking the top 100 items. Furthermore, compared with DRN prediction with re-ranking all 200 items, DDRN is 6.94 times faster than DRN + MMR in TP999. This is mainly because DRN+re ranking models are a serial process of ranking first and then re-ranking, while DDRN is a single parallel ranking process that costs less time. These results demonstrate that improving diversity in the ranking stage is more efficient than the one in the re-ranking stage.

**Table 6: Time performance comparison**

| Model | Top-$k$ | TP999 (ms) | TP50 (ms) | MEAN (ms) |
|---|---|---|---|---|
| DRN+MMR | 200 | 295.98 | 274.87 | 282.24 |
| | 150 | 138.95 | 128.23 | 129.02 |
| | 100 | 68.89 | 56.43 | 58.27 |
| DRN+CATE | 200 | 153.89 | 141.56 | 142.13 |
| | 150 | 119.25 | 104.87 | 105.63 |
| | 100 | 58.29 | 51.02 | 51.16 |
| DRN+DPP | 200 | 110.67 | 101.69 | 102.62 |
| | 150 | 92.56 | 82.88 | 84.28 |
| | 100 | 47.89 | 38.99 | 40.81 |
| DDRN | – | **37.27** | **33.6** | **34.16** |

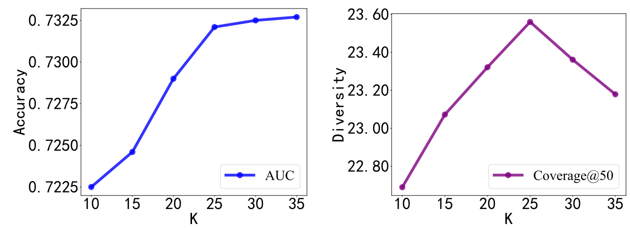
## 4.8 Hyper-parameter Tuning of DDRN (RQ7)

In this section, we study the influence of hyper-parameters of our Diversity-aware Deep Ranking Network (DDRN). In DDRN, there are two hyper-parameters ($K$ and $\gamma$) that heavily affect the model performance. $K$ is the number of items selected from the embedding sequence $S$ in the diversity-aware listwise information fusion module, and $\gamma$ is a parameter defining the maximal weight in the balanced weighting loss.

We report the hyper-parameter tuning results on Meituan dataset as a representative. We use AUC as the accuracy metric and Coverage50 as the diversity metric. Parameter $K$ is tuned in range of [10, 15, 20, 25, 30, 35], and parameter $\gamma$ tuned in the range of [1, 2, 3, 4, 5, 6]. Figure 5 and Figure 6 visualize the tuning results of $K$ and $\gamma$, respectively.

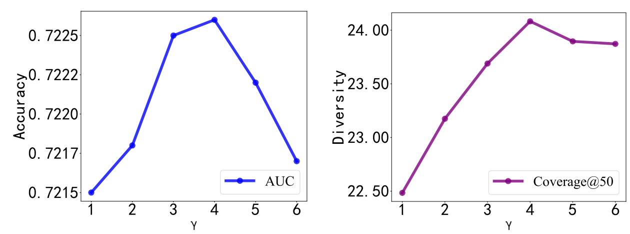
In Figure 5, when $K$ is increased from 10 to 25, the AUC and Coverage keep rising which implies that the filtered embedding sequence improves both the accuracy and diversity of recommendation. However, when $K$ is further increased to 35, the AUC is only improved a little, and the Coverage is decreased significantly. This is because the larger $K$ brings more unimportant items, making the filtered sequence between items less different, and weakening the ability of the model to distinguish similar items.

In Figure 6, when $\gamma$ is increased from 1 to 4, the AUC and Coverage are improved by balancing the biased gradients between popular categories and unpopular categories (i.e., the categories with few interactions). When $\gamma$ is further increased to 6, the Coverage are stable and the AUC is decreased. This is because the unpopular categories are weighted excessively, which breaks the gradient balance between popular categories and unpopular categories again.




(a) Accuracy (b) Diversity

**Figure 5: Influence of parameter $K$.**




(a) Accuracy (b) Diversity

**Figure 6: Influence of parameter $\gamma$.**

## 5 CONCLUSION

In this work, we proposed a Diversity-aware Deep Ranking Network to improve recommendation diversity in the ranking stage. DDRN not only preserves the diversity of the matching stage, but also provides a diversified list for the re-ranking stage. DDRN alleviates the limitation of pointwise ranking model and long-tail distribution of user interaction history on recommendation diversity through two careful diversity-aware designs. We also conducted comprehensive experiments on both offline and online environment and experimental results clearly demonstrated that DDRN can effectively improve the diversity of recommendation in the ranking stage, while maintaining the accuracy.

## ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Nos. 62272054, 62192784) and Xiaomi Young Talents Program.

## REFERENCES


[1] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. 2015. Optimal greedy diversity for recommendation. In *IJCAI*.

[2] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*. 335–336.

[3] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable Multi-Interest Framework for Recommendation. In *SIGKDD*. 2942–2951.

[4] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–28.

[5] Laming Chen, Guoxin Zhang, and Eric Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems* 31 (2018).

[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *RecSys*. 191–198.

[7] Qinxu Ding, Yong Liu, Chunyan Miao, Fei Cheng, and Haihong Tang. 2021. A hybrid bandit framework for diversified recommendation. In *AAAI*, Vol. 35. 4036–4044.

[8] Lu Gan, Diana Nurbakova, Léa Laporte, and Sylvie Calabretto. 2020. Enhancing recommendation diversity using determinantal point processes on knowledge graphs. In *SIGIR*. 2001–2004.

[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[10] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[11] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *AAAI*, Vol. 30.

[12] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. 355–364.

[13] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*. 2333–2338.

[14] Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems–A survey. *Knowledge-based systems* 123 (2017), 154–162.

[15] Chang Li, Haoyun Feng, and Maarten de Rijke. 2020. Cascading hybrid bandits: Online learning to rank for relevance and diversity. In *RecSys*. 33–42.

[16] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *CIKM*. 2615–2623.

[17] Zhi Li, Jinshan Yang, Xu Gou, and Xiaorong Qi. 2019. Recurrent neural networks with segment attention and entity description for relation extraction from clinical texts. *Artificial intelligence in medicine* 97 (2019), 9–18.

[18] Yile Liang, Tieyun Qian, Qing Li, and Hongzhi Yin. 2021. Enhancing domain-level and user-level adaptivity in diversified recommendation. In *SIGIR*. 747–756.

[19] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[20] Yong Liu, Yingtai Xiao, Qiong Wu, Chunyan Miao, Juyong Zhang, Binqiang Zhao, and Haihong Tang. 2020. Diversified interactive recommendation with implicit feedback. In *AAAI*, Vol. 34. 4932–4939.

[21] Zheng Liu, Jianxun Lian, Junhan Yang, Defu Lian, and Xing Xie. 2020. Octopus: Comprehensive and elastic user representation for the generation of recommendation candidates. In *SIGIR*. 289–298.

[22] Ze Lyu, Yu Dong, Chengfu Huo, and Weijun Ren. 2020. Deep match to rank model for personalized click-through rate prediction. In *AAAI*, Vol. 34. 156–163.

[23] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *SIGKDD*. 1930–1939.

[24] S Patro and Kishore Kumar Sahu. 2015. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462* (2015).

[25] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.

[26] Shameem A Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. 2016. A coverage-based approach to recommendation diversity on similarity graph. In *RecSys*. 15–22.

[27] Steffen Rendle. 2010. Factorization machines. In *ICDM*. IEEE, 995–1000.

[28] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*. 175–186.

[29] Chaofeng Sha, Xiaowei Wu, and Junyu Niu. 2016. A framework for recommending relevant and diverse items.. In *IJCAI*, Vol. 16. 3868–3874.

[30] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-interest network for sequential recommendation. In *WSDM*. 598–606.

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

[32] Romain Warlop, Jérémie Mary, and Mike Gartrell. 2019. Tensorized determinantal point processes for recommendation. In *SIGKDD*. 1605–1615.

[33] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: neural news recommendation with personalized attention. In *SIGKDD*. 2576–2584.

[34] Chuhan Wu, Fangzhao Wu, Mingxiao An, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with topic-aware news representation. In *ACL*. 1154–1159.

[35] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. 2019. PD-GAN: Adversarial Learning for Personalized Diversity-Promoting Recommendation.. In *IJCAI*, Vol. 19. 3870–3876.

[36] Qiong Wu, Yong Liu, Chunyan Miao, Yin Zhao, Lu Guan, and Haihong Tang. 2019. Recent advances in diversified recommendation. *arXiv preprint arXiv:1905.06589* (2019).

[37] Le Yan, Zhen Qin, Rama Kumar Pasumarthi, Xuanhui Wang, and Michael Bendersky. 2021. Diversification-aware learning to rank using distributed representation. In *Proceedings of the Web Conference 2021*. 127–136.

[38] Mi Zhang and Neil Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys*. 123–130.

[39] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. 2021. DGCN: Diversified Recommendation with Graph Convolutional Networks. In *WWW*. 401–412.

[40] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *SIGKDD*. 1059–1068.

[41] Han Zhu, Junqi Jin, Chang Tan, Fei Pan, Yifan Zeng, Han Li, and Kun Gai. 2017. Optimized cost per click in taobao display advertising. In *SIGKDD*. 2191–2200.