A Bird’s-eye View of Reranking: from List Level to Page Level

Yunjia Xi∗  
xiyunjia@sjtu.edu.cn  
Shanghai Jiao Tong University  
Shanghai, China

Jianghao Lin†  
chiangel@sjtu.edu.cn  
Shanghai Jiao Tong University  
Shanghai, China

Weichen Liu‡  
liuweichen@huawei.com  
Huawei Noah’s Ark Lab  
Shenzhen, China

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

Weinan Zhang  
wznzhang@sjtu.edu.cn  
Shanghai Jiao Tong University  
Shanghai, China

Yong Yu  
yyu@sjtu.edu.cn  
Shanghai Jiao Tong University  
Shanghai, China

Ruiming Tang  
tangruiming@huawei.com  
Huawei Noah’s Ark Lab  
Shenzhen, China

ABSTRACT
Reranking, as the final stage of multi-stage recommender systems, refines the initial lists to maximize the total utility. With the development of multimedia and user interface design, the recommendation page has evolved to a multi-list style. Separately employing traditional list-level reranking methods for different lists overlooks the inter-list interactions and the effect of different page formats, thus yielding suboptimal reranking performance. Moreover, simply applying a shared network for all the lists fails to capture the distinctions in user behaviors on different lists. To this end, we propose to draw a bird’s-eye view of page-level reranking and design a novel Page-level Attentional Reranking (PAR) model. We introduce a hierarchical dual-side attention module to extract personalized intra- and inter-list interactions. A spatial-scaled attention network is devised to integrate the spatial relationship into pairwise item interactions, which explicitly models the page format. The multi-gated mixture-of-experts module is further applied to capture the commonalities and differences of user behaviors between different lists. Extensive experiments on a public dataset and a proprietary dataset show that PAR significantly outperforms existing baseline models.

CCS CONCEPTS
• Information systems → Recommender systems.

KEYWORDS
Reranking, Recommender System, Multi-block Page

∗Both authors contributed equally to this research.
†The corresponding author.
‡Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Yunjia Xi, Jianghao Lin, Weichen Liu, Xinyi Dai, Weinan Zhang, Rui Zhang, Ruiming Tang, and Yong Yu. 2023. A Bird’s-eye View of Reranking: from List Level to Page Level. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM ’23), February 27–March 3, 2023, Singapore, Singapore. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570399

1 INTRODUCTION
In multi-stage recommender systems (MRS), reranking, as the final stage, re-orders the input ranking lists from the previous ranking stage [21]. The goal is to maximize the total utility of the reranked lists. The quality of reranking has a direct impact on users’ experience and satisfaction, and thus plays a crucial role in MRS [26].

Various reranking methods [2, 3, 32, 33, 44] have been developed in recent years, but they are mainly list-level reranking models. List-level models rerank a single list each time, and only consider the cross-item influence within the individual list [26]. Such a reranking strategy, though already found useful in many industrial applications, may still be suboptimal. In fact, with the development of multimedia and user interface design, the final recommendation page presented to the user is usually in a multi-list style [14, 15]. As shown in Figure 1, each list on the multi-list page often highlights a particular theme (e.g., “Trending,” “Games”), sometimes even with a tailored layout (e.g., size, location). The existence of other lists changes the user behavior patterns, leading to a different utility distribution. In this work, we propose to draw a bird’s-eye view over the whole reranking page and develop a page-level reranking algorithm. While intuitively useful to integrate page-wise information, it is nontrivial to jointly perform reranking for multiple lists with the following three major challenges (C1 - C3).

(C1) Firstly, incorporating intra-list and inter-list interactions is essential for page-level reranking. Items from the same list are usually related to the same theme. Modeling the cross-item influence within the list (i.e., intra-list interaction) and identifying the best permutation over candidate items is the main objective for reranking [26]. Moreover, we observe that the inter-list interaction is also crucial for page-level modeling — whether a user is interested in an item is also influenced by items placed in other lists. Independent
To address the above issues, we propose a novel model named Page-level Attentional Reranking (PAR) for page-level reranking. Multiple lists are jointly reranked with a unified model to capture the multifaceted fine-grained mutual influences among lists. Firstly, we propose hierarchical dual-side attention (HDS-Attn) module to extract the intra- and inter-list interactions according to users’ individual behavior history (C1). Next, the spatial-scaled attention (SS-Attn) network is designed to encapsulate the pairwise influence between items with respect to their spatial relationship. The attention is numerically scaled by the distance between items on the page, which provides explicit modelings over the page format (C2). Lastly, after obtaining the interacted feature representation from HDS-Attn and SS-Attn, PAR adopts the Multi-gated Mixture-of-Experts (MMoE [30]) module to capture the commonalities and differences of user behavior patterns among different lists (C3). MMoE maintains a set of parallel expert networks to capture various aspects of behavior patterns and applies an attentional gate with list-specific parameters to aggregate the expert outputs for the final score estimation. The main contributions of this paper are:

- We propose to draw a bird’s-eye view over the whole reranking page to jointly rerank multiple lists and optimize the overall utility by considering the page-wise information. To the best of our knowledge, this is the first work to consider the effect of general page format for page-level reranking.
- We conduct data analysis based on a multi-list dataset and identify the importance of intra- and inter-list interactions and the spatial relationship (i.e., reflection of page format) between items.
- We propose a novel Page-level Attentional Reranking (PAR) model. We design an HDS-Attn module for the personalized intra- and inter-list interactions, and an SS-Attn module to incorporate the page format. The MMoE module is employed to capture the commonalities and distinctions for different lists.
- Extensive experiments on a public dataset and a proprietary dataset show that PAR achieves the state-of-the-art performance compared with existing baseline models.

2 DATA ANALYSIS

One of the challenges for page-level reranking is to model the intra- and inter-list interactions between items on a page. Therefore, we present a brief data-driven study on an F-shape page (i.e., Figure 2a) to investigate how items at different positions influence each other. The data is collected from a mainstream App Store, where the page adopts an interleaved arrangement of vertical and horizontal lists. The detailed data description can be found in Section 4.1.1.

To study how the other items placed on the same page influence the utility of a given item, we select 6 fixed positions on the F-shape page, as illustrated in Figure 2a. For the fixed position $P_1$:

- $P_2$ is adjacent to $P_1$ in the same list;
- $P_3$ is distant from $P_1$ in the same list;
- $P_4$ is adjacent to $P_1$ in another neighboring list;
- $P_5$ is distant from $P_1$ in another neighboring list;
- $P_6$ is distant from $P_1$ in a remote list.

They form five typical positional relationships on a multi-list page. For each pair of positions (i.e., $P_i$ versus $P_j$, $i, j = 2, 3, 4, 5, 6$), we compute the click-through rate (CTR) of $P_i$ with different categories when the item category at $P_j$ varies. We plot the heatmap for the
A page-level reranking model takes as inputs the multiple ordered initial lists on the same page generated by the previous rankers, and refines the ranking lists by considering the mutual influence between items and the impact of the page format.
3.4.1 Hierarchical Dual-side Attention. One of the challenges of page-level reranking is to incorporate intra- and inter-list interactions on a page (C1 in Section 1). Since there is a natural hierarchical structure of items forming a list and lists forming a page, we propose to consider intra-list and inter-list interaction in a hierarchical way. As shown in the left part of Figure 3, the IIDS-Attn module consists of dual-side attention, item-level aggregation, list-level self-attention, and list-level aggregation, from bottom to top. The bottom two parts are designed to extract the intra-list interaction within each list. To provide personalized reranking, the cross-item influence on the candidate item side requires the personal preference information from the history list side. Therefore, we introduce the dual-side attention for each list to model the mutual influence between these two sides. Then, item-level aggregation is employed to combine the item information within a list and generate list representation. The upper two parts, list-level self-attention and list-level aggregation, are designed to capture the inter-list interaction between lists and generate the final page representation.

Dual-side Attention. Users’ history list carries rich information for inferring their personal preferences and tastes, which is helpful for reranking [24, 40]. Moreover, the items in the users’ history list contribute differently for different candidate lists [34, 40]. For example, game apps in the user’s history list may be more critical when reranking the “Top-10 Games” list for the user. Inspired by [29], we design the dual-side attention to model the fine-grained correlations between the candidate item side and history list side.

Formally, for each list \( R_i \) on page \( P \), \( i = 1, \ldots, n \), the input of the dual-side attention is the corresponding \( i \)-th item embedding matrix \( \tilde{X}_i \in \mathbb{R}^{m \times d_x} \) of the candidate list \( R_i \) and the history embedding matrix \( \tilde{H} \in \mathbb{R}^{l \times d_h} \). We maintain an affinity matrix \( W^i \in \mathbb{R}^{d_h \times d_x} \) for each list to learn the importance of the association between each pair of items from both candidate and history sides:

\[
\begin{align*}
C_i &= \tanh \left( \tilde{H} W^i \tilde{X}_i \right), \\
A^s_i &= \text{Softmax} \left( \tanh \left( \tilde{X}_i W^s \right) + \left( \tilde{H} W^s \right)^\top C_i \right), \\
A^h_i &= \text{Softmax} \left( \tanh \left( \tilde{H} W^h \right) \tilde{X}_i W^h \right)^\top. 
\end{align*}
\]

where \( W^s \in \mathbb{R}^{d_x \times m}, W^h \in \mathbb{R}^{d_h \times m} \) are learnable weight matrices. The matrices \( A^s_i \in \mathbb{R}^{m \times m} \) and \( A^h_i \in \mathbb{R}^{m \times l} \) after the softmax function represent the attention weights of items in the candidate list and history list. Then the interacted representation matrices \( \tilde{X}_i = [\tilde{x}_{i,j}]_{j=1}^{m} \in \mathbb{R}^{m \times d_x} \) and \( \tilde{H}_i = [\tilde{h}_{i,j}]_{j=1}^{l} \in \mathbb{R}^{m \times d_h} \) now contain useful information from both candidate list \( R_i \) and history list \( H_i \).

Item-level Aggregation. We apply the item-level aggregation to learn the intra-list interaction and generate the list representations. Since items contribute differently to the representation of the target list \( R_i \), we aggregate the attained item representations with an attention mechanism [41] to form the list representation \( l_i \):

\[
\begin{align*}
  u_{i,j} &= \tanh \left( W^i_j [\tilde{x}_{i,j} \parallel \tilde{h}_{i,j}] + b_i \right), \\
  a_{i,j} &= \frac{\exp(u^i_{j,\text{Item}})}{\sum_{j=1}^{m} \exp(u^i_{j,\text{Item}})}, \\
  l_i &= \sum_{j=1}^{m} a_{i,j} [\tilde{x}_{i,j} \parallel \tilde{h}_{i,j}],
\end{align*}
\]

where \( \parallel \) denotes the vector concatenation. We first feed the concatenated item representations into a linear layer to get \( u_{i,j} \) for each item \( x_{i,j} \). Then, \( W^i_j \in \mathbb{R}^{d_x \times d_x} \) and \( b_i \in \mathbb{R}^d \) are the learnable weights. Then the importance of each item is measured by the similarity of \( u_{i,j} \) with a item-level query vector \( q_{\text{Item}} \in \mathbb{R}^d \). The item-level query vector \( q_{\text{Item}} \) is a trainable parameter and serves as the attention query in the item-level aggregation. Next, we normalize the weights \( a_{i,j} \) and compute the list representation \( l_i \) by the weighted sum of each item. To this end, important intra-list interaction has been fused into the list representation \( l_i \). Stacking all the list representations, we get the list representation matrix \( L = [l_i]_{i=1}^n \in \mathbb{R}^{n \times d} \) for further use.

List-level Self-attention. Given the list representation matrix \( L \), we model the inter-list influence between different lists on the page through a list-level self-attention layer:

\[
\tilde{L} = \text{Softmax}(LL^\top / \sqrt{d})L,
\]

where \( \tilde{L} = [\tilde{l}_i]_{i=1}^n \in \mathbb{R}^{n \times d} \) is the re-weighted list representation matrix that captures the relationship and correlations between different lists, and \( \sqrt{d} \) is used to stabilize gradients during training.

List-level Aggregation. Finally, built upon the re-weighted list representations \( \tilde{L} \), we employ the list-level aggregation layer to combine the information from different lists and generate the unified page representation \( S \). Similar to the item-level aggregation, after a linear transformation, we involve a learnable list-level query vector \( q_{\text{List}} \in \mathbb{R}^d \) for calculating the attention weights \( \beta \). Then the re-weighted list representations \( \tilde{L} \) are aggregated into a shared page representation.
representation $S$ by a weighted sum:

$$v_i = \tanh (W_p \hat{l}_i + b_p),$$

$$\hat{p}_i = \frac{\exp(v_i^T q_{list})}{\sum_{i'=1}^n \exp(v_i^T q_{list})},$$

$$S = \sum_{i=1}^n \hat{p}_i \hat{l}_i,$$

where page representation $S$ integrates the information of page-wise contexts and history behaviors, and is shared for all the lists.

### 3.4.2 Spatial-scaled Attention

With multiple lists on one page, the arrangement of lists and items, i.e., the page format, becomes an essential issue to be considered (C2 in Section 1). However, no previous work on page-level reranking has discussed such an issue before. Different page formats change the location of items and therefore change the distances between them. In Section 2, we observe that the influence between items generally shows a negative correlation with the distance between them. Hence different page formats yield diverse influences between items. Thus, we propose the SS-Attn module to estimate the pairwise item influence with the consideration of the spatial relationship. Specifically, SS-Attn adjusts the attention weights according to the relative distance between items on the page, so that closer items have a stronger influence on each other. The relative distance can be altered depending on the page format, showing that the SS-Attn is flexible and can be adapted to different formats.

As such, with a total of $nm$ items on the page, we introduce a symmetric distance matrix $D \in \mathbb{R}^{nm \times nm}$, whose element $d_{p,q} \geq 0$ indicates the geometric distance between the corresponding pair of items $(p,q)$. In this work, we adopt the Manhattan distance to build the distance matrix. For example, in Figure 2a, the Manhattan distances from $P_1$ to $P_7$ $(i = 2,3,4,5,6)$ are 1, 2, 1, 2, 5, respectively. It is worth noting that the design of the distance matrix is flexible and can be customized according to the page formats in different page-level reranking scenarios. Other distance measurements (e.g., Euclidean distance, slot counting) are also applicable.

Concretely, we first reshape the embedding matrix of the candidate items $X \in \mathbb{R}^{n \times m \times d}$ into $X \in \mathbb{R}^{n \times m \times d_x}$ where each row $x_p$ $(p = 1, \ldots, nm)$ of the matrix $X$ is the feature vector of a candidate item. The input of the SS-Attn module is the reshaped item matrix $X$ and the distance matrix $D$. To involve the page format effect (i.e., the larger the distance, the less the influence between items), we transform the distance matrix by a learnable sigmoid function [38].

The learnable sigmoid function $f$ parameterized by a scalar $\sigma$ maps the distance $d_{p,q}$ of item pair $(p,q)$ to a positive distance-aware influence factor $\hat{d}_{p,q}$ of range $(0,1]$, $f(\cdot) : \mathbb{R} \rightarrow (0,1]$.

$$\hat{d}_{p,q} = f(d_{p,q}|\sigma) = \frac{1 + \exp(\sigma)}{1 + \exp(\sigma d_{p,q})},$$

where $\sigma \in \mathbb{R}$ is a learnable scalar that determines the steepness of function $f(\cdot)$, and $\sigma > 0$ is a hyper-parameter for normalizing the distance $d_{p,q}$ and stabilizing the training. In our experiment, $\sigma$ is set to 0.1. Note that $f(\cdot)$ is a monotonically decreasing function w.r.t. the distance and satisfies $f(0) = 1$ and $f(+\infty) = 0$, i.e., the influence of between the items gradually decreases from 1 to 0 as the distance grows.

Next, we use the obtained influence factor $\hat{d}_{p,q}$ to scale the pairwise mutual influence between items on the page. Multi-head attention is adopted for modeling the interactions between any pair of items on the page, while the attention weights are scaled according to the distance-aware influence factor. Suppose $B$ is the number of heads, we maintain $B$ different learnable sigmoid functions to learn different levels of the page format effect, with an individual parameter $\hat{\sigma}^{(b)}$ for the $b$-th attention head, $b = 1, \ldots, B$. We form all the $\hat{d}_{p,q}$ for the $b$-th head into a matrix $\hat{D}^{(b)}$, and numerically scale the pairwise self-attention weights:

$$\hat{O}^{(b)} = \text{Softmax} \left( \frac{\phi((XW_Q^{(b)})(XW_K^{(b)})^T) \odot \hat{D}^{(b)}}{\sqrt{d_a}} \right) (XW_V^{(b)}),$$

where $\odot$ is the element-wise product, the preliminary attention weights $(XW_Q^{(b)})(XW_K^{(b)})^T$ are adjusted according to the distance-aware influence factors, and $d_a$ is the dimension of the vectors in $XW_Q^{(b)}$ and $XW_K^{(b)}$. Note that the non-negative monotonically increasing function $\phi$ is introduced to avoid negative attention weights, as negative preliminary attention weights can invert the distance-aware influence and violates the negative correlation between distances and influences. Here, we use softplus function.

Finally, we concatenate the multi-head spatial-scaled attention outputs, and apply a linear transformation to get the pairwise influence matrix $\hat{O} \in \mathbb{R}^{nm \times d_a} = \hat{O}^{(1)}$ || $\ldots$ || $\hat{O}^{(B)} W_Q$, where $d_a$ is the attention output size. We reshape the matrix $\hat{O} \in \mathbb{R}^{nm \times d_a}$ back to $\hat{O} \in \mathbb{R}^{n \times m \times d_a}$, where the vector $a_{i,j} \in \mathbb{R}^{d_a}$ denotes the pairwise influence vector for the $j$-th item in the $i$-th list.

### 3.4.3 Dense Network

In addition to the HDS-Attn and SS-Attn, we employ a fully-connected network to capture the implicit feature interactions within each item. We feed each item embedding $\hat{x}_{i,j}$ into a shared MLP to obtain the dense feature $r_{i,j} = \text{MLP}(\hat{x}_{i,j})$ for the latter reranking score estimation.

### 3.5 Reranking Score Estimation

After the page-level interaction layer, we obtain the shared page representation $S$, the pairwise item influence vector $o_{i,j}$, and the dense feature $r_{i,j}$ for item $x_{i,j}$. Although these features incorporate page and item-level information, the commonalities and distinctions of user behaviors on different lists are remained to be solved.

As discussed before (C3 in Section 1), the user’s behaviors may not only share some basic patterns (e.g., position bias) and underlying preferences, but also have distinctions for different lists due to themes and formats. To this end, we adopt the Multi-gated Mixture-of-Experts (MMeE [30]) module, where several expert sub-models are shared across all lists. Each list possesses a specific gating network to ‘select’ a subset of experts to use. Through the expert and gating networks, our model automatically adjusts parameterization between modeling shared information and list-specific information, so as to exploit the common behavior patterns while paying attention to the list-specific patterns.

As shown in Figure 3, there are $E$ parallel expert networks $(g_{i,k})_{k=1}^E$, which are all MLPs with ReLU activations, to capture different aspects of behavior patterns. For each list $i$, we maintain a separate fully-connected gating network $g_i(\cdot)$ to learn a linear
We randomly sample lists from all the themes if less than four lists with at least one click. Next, each page is formed by four lists from we first construct lists from her/his positively interacted themes of the public CTC dataset to construct the pages. For each user, different themes that are horizontally stacked from top to bottom.

\[ y_{i,j} = \text{Softmax}(g_i(\{S \parallel r_i \parallel o_{i,j}\})) \]

\[ \hat{y}_{i,j} = \frac{t_i(\hat{z}_{i,j})}{\sum_{k=1}^{E} y_{i,j,k} \times e_k(\{S \parallel r_i \parallel o_{i,j}\})} \]

We sort items in each list by the scores \( \hat{y}_{i,j} \) to get the final rerankings. Given the click label matrix \( Y \) of size \( n \times m \) where \( y_{i,j} \) denotes the click signal for the \( j \)-th item in the \( i \)-th list, we optimize the model via binary cross-entropy loss on the training page set \( P \):

\[ L = \sum_{\mathcal{P}} \sum_{i=1}^{n} \sum_{j=1}^{m} y_{i,j} \log \hat{y}_{i,j} + (1 - y_{i,j}) \log(1 - \hat{y}_{i,j}) \] (7)

**Computational Complexity Analysis.** The complexity of PAR is \( O(n^2m^2) \), where \( n \) is the number of lists and \( m \) is the length of the lists. Its complexity is comparable to most existing models of \( O(nm^3) \) (e.g., PRM [33], DHANR [9]), as the number of lists on the page \( n \) is usually less than 5 due to the limit of the screen size.

## 4 EXPERIMENT

### 4.1 Experiment Settings

#### 4.1.1 Datasets.

Our experiments are conducted on a public dataset, Cloud Theme Click Dataset\(^1\) and a proprietary dataset, AppStore.

- **Cloud Theme Click Dataset** [10] (CTC for short) records the click data of Cloud Theme in Taobao app. The dataset includes 1,423,835 click records from 355 different themes during a 6-day promotion season, users’ purchase history before the promotion, and the embedding of 720,210 users and 1,361,672 items.

- **AppStore** is collected from a mainstream commercial app store, from October 16, 2021 to November 1, 2021. The dataset contains 47,003,121 pages, 28,632,998 users and 1365 apps. Each app has 32 features (e.g., app developer, app category). Each page is in the form of an F-shape with one vertical list inserted by four horizontal lists, and each user has a history list of behaviors collected in real time.

#### 4.1.2 Page and click generation for public dataset. As there is no publicly available dataset with page data, we use the click logs of the public CTC dataset to construct the pages. For each user, we first construct lists from her/his positively interacted themes with at least one click. Next, each page is formed by four lists from different themes that are horizontally stacked from top to bottom. We randomly sample lists from all the themes if less than four lists have been clicked. Four DNN models are trained separately as the initial rankers to generate the initial rankings of length 10.

An oracle click model (e.g., [8, 25]) is then adopted to simulate necessary click data on the obtained new pages. We mainly follow the click simulation in Seq2Slate [4] to decompose the click probability of an item into the product of relevance, position decaying, and dissimilarity probability. We use the original click label of an item as the relevance probability, which is equal to 1 if clicked, and 0 otherwise. If the item is placed at \( i \)-th position in the \( j \)-th list, the position decaying probability is \( 1/(i^{1/p}) \), where \( p_1 \) and \( p_2 \) are the horizontal and vertical decay parameters. Here we set \( p_1 = 2 \) to 0.4 and 0.5. When observing an item, the user may tend to click the item dissimilar/diverse to its surrounding items [18, 28]. The cosine similarity between item embeddings is used to compute the dissimilarity probability to introduce high-order interaction between items. After the generation of clicks, all baselines and our model PAR are trained based on the synthetic click data.

**4.1.3 Baselines.** Since currently there is only one work, DHANR [19], that focuses on page-level reranking, we modify some models in related fields that utilize page-level information as baselines, e.g., HMoE [23] in multi-scenario ranking, TRNN [27] in whole-page optimization. We also design GlobAtt to a global multi-head self-attention structure to model the mutual influence between any pairs of items on the page. We also choose miDNN [44], GSF [3], DLCM [2], PRM [33], and SetRank [32] as list-level reranking baselines for multiple lists on the same page. Multiple models are trained separately on different lists.

#### 4.1.4 Evaluation metrics.

All the reranking models are evaluated in terms of relevance-based and utility-based metrics. For relevance-based metrics, we adopt widely-used MAP and nDCG [22] following previous work [2, 27, 33]. Despite that there are multiple lists on the page, we calculate nDCG and MAP for each list and report their averaged nDCG and MAP.

As for utility-based metrics, we employ the average number of clicks on the page Utility, and the sum of the click probabilities for all items on a page \( sCTR \), following [39]. In addition, we also compute the sum of click probabilities on each list, such as the sum of click probabilities on vertical lists \( sCTR_{vi} \) and the sum of click probabilities on the first horizontal lists \( sCTR_{hi} \). On the public CTC dataset, the click probabilities and clicks of the reranked lists are generated by the same oracle click model used in 4.1.2. The AppStore dataset records the real user clicks on the F-shape page. Its click probabilities and clicks on the reranked lists are given by a click model for the F-shape page. FSCM [14].

#### 4.1.5 Reproducibility.

The implementation of our model is available\(^2\). We adopt Adam as the optimizer. The learning rate is \( 2 \times 10^{-4} \), and the parameter of \( l_2 \) regularization is \( 2 \times 10^{-4} \). The batch size and the embedding size of the categorical feature are set to 128 and 16. The number of experts and the architecture of experts and towers in MMoE are 12, [200, 80], and [80], respectively. To ensure a fair comparison, we also fine-tune all baseline models to achieve their best performance.

### 4.2 Overall Performance

The overall performance is reported in Table 1, from which we have the following observations.

Firstly, our model PAR performs significantly better than all the baselines on both datasets. Methods with page-wise information generally work better than list-level methods, validating the benefit of utilizing page-level information. PAR surpasses all the baselines on the two datasets. As presented in Table 1, PAR improves over the best baseline on CTC dataset with respect to Utility, sCTR, nDCG, and MAP by 5.276%, 3.115%, 4.623%, and 8.485%, respectively. On\(^3\)

---

\(^1\)https://tianchi.aliyun.com/dataset/dataDetail?dataId=9716

\(^2\)The TensorFlow implementation is available at: https://github.com/YunjiaXi/Page-level-Attentional-Reranking. The MindSpore implementation is available at: https://gitee.com/mindspore/models/tree/master/research/recommend/PAR

\(^3\)The MindSpore implementation is available at: https://github.com/YunjiaXi/Page-level-Attentional-Reranking. The MindSpore implementation is available at: https://gitee.com/mindspore/models/tree/master/research/recommend/PAR
AppStore dataset, PAR also achieves 6.432% and 4.217% improvement over the best baseline in terms of Utility and sCTR. This demonstrates the necessity of modeling the multifaceted dynamic interactions and the page format in page-level reranking.

Secondly, different page formats result in different click distributions. As illustrated in Table 1, the top clicks are predominantly located on the vertical and the first horizontal lists on the F-shape pages of AppStore dataset. On the all-row pages of CTC dataset, clicks are concentrated on the first three horizontal lists and the probability of clicking decreases with the positions of the horizontal list. PAR shows greater improvement in these major lists by exploiting more useful information. Page-level baselines outperform list-level models on AppStore dataset, but sometimes this is not the case for the CTC dataset, which may also be due to the different page formats. The inter-list interaction for pages of multiple horizontal lists may be less than that for F-shape pages.

Lastly, the performance of the models on the AppStore and CTC datasets diverges in terms of the relevance-based metrics, nDCG and MAP. On the CTC dataset, PAR achieves the best nDCG and MAP, but on the AppStore dataset, the best relevance-based metrics are achieved by the list-level reranking method, PRM. This may be attributed to the fact that CTC dataset has groundtrue relevance labels, whereas the AppStore does not. The click labels in AppStore dataset are directly used for computing the relevance-based metrics, following [2, 27, 33]. Yet the clicks could be biased, and there exist some relevant items that have not been clicked. The list-level approaches tend to place the past-clicked items first, and therefore obtain higher nDCG and MAP. In comparison, the page-level approaches combine information from multiple lists to find relevant items that might not have been clicked to optimize the total utility and often do not necessarily place clicked items at top positions. Such an observation shows that using past clicks for evaluation may not be able to reflect the true performance for page-level reranking.

### 4.3 In-depth Analysis

#### 4.3.1 Ablation Study

To investigate the effectiveness of each component in PAR, we design several variants of PAR and conduct a series of experiments on AppStore and CTC datasets.

- **PAR-DASA** replaces the dual-side attention in HDS-Attn with a self-attention, thus removing the user history from the module.
- **PAR-HDSSA** removes the HDS-Attn module.
- **PAR-scale** replaces the SS-Attn module with self-attention.
- **PAR-SSA** removes the SS-Attn module.
- **PAR-DN** removes the dense network.
- **PAR-MMoE** replaces the MMoE module with a single MLP.

![Figure 4: (a) The attention weights (normalized along the horizontal axis) of the list-level self-attention in HDS-Attn. (b) The pairwise attention weights between the target item (highlighted by a red box) and other items in SS-Attn.](image)

The comparison of the above variants and the original PAR on AppStore and CTC datasets are presented in Table 2. Compared to the original PAR, the performances of the variants all decline to some extent, indicating the effectiveness of each module. Among all the variants, PAR-HDSSA generally suffered the greatest drop in utility, which suggests that incorporating intra- and inter-list interaction can enhance the performance of page-level reranking. The decline of PAR-DASA indicates the importance of personalized preferences in user behaviors for reranking. Removing the MMoE and SSA modules also introduces a large decrease, illustrating the impact of modeling the differences and commonalities between lists and the page format. There is no significant gap between the results of PAR-SSA and PAR-scale, revealing that it is the distance-aware influence factor that contributes primarily to the SSA module. The basic self-attention is insufficient to model different page formats.

#### 4.3.2 Case Study

To explore the mutual influence between lists and the spatial relationship between items, we select a page from the AppStore and visualize the attention weights from list-level self-attention in HDS-Attn and spatial-scaled attention (SS-Attn).

In Figure 4(a), all lists have a high attention weight for the AppStore and visualize the attention weights from list-level self-attention in HDS-Attn. In Figure 4(b), the pairwise attention weights between the target item (highlighted by a red box) and other items in SS-Attn.

The comparison of the above variants and the original PAR on AppStore and CTC datasets is presented in Table 2. Compared to the original PAR, the performances of the variants all decline to some extent, indicating the effectiveness of each module. Among all the variants, PAR-HDSSA generally suffered the greatest drop in utility, which suggests that incorporating intra- and inter-list interaction can enhance the performance of page-level reranking. The decline of PAR-DASA indicates the importance of personalized preferences in user behaviors for reranking. Removing the MMoE and SSA modules also introduces a large decrease, illustrating the impact of modeling the differences and commonalities between lists and the page format. There is no significant gap between the results of PAR-SSA and PAR-scale, revealing that it is the distance-aware influence factor that contributes primarily to the SSA module. The basic self-attention is insufficient to model different page formats.
its surrounding items roughly follow the pattern that the further away the item is, the less weight it gets. Furthermore, we find that there are outliers that violate the spatial decay pattern. In the first horizontal list, the first item $i_1$ and second item $i_2$ have higher attention weights $(0.0149$ and $0.0150)$ than the third item $i_3$ $(0.0148)$, though $i_3$ is closer to the target item. The possible reason is that $i_1$ and $i_2$ share the same category with the target item (they are all short-video apps), while $i_3$ is a search engine app. The similar category promotes the pairwise mutual influence between items. As such, we conclude that the SS-Attn can automatically learn the combination of the spatial effect and the pairwise influence.

5 RELATED WORKS

5.1 Reranking

Most existing reranking methods are list-level and rerank separately for each individual list [2, 4, 12, 13, 32, 33, 39]. Various network structures have been applied for modeling the mutual influences within the list. For example, miDNN [44] uses DNN with global feature extension to capture mutual influences between items. A group-wise scoring function (GSF) [3] is learned by enumerating all the feasible item permutations of the list. DLCM [2] employs the gated recurrent unit (GRU) to encode the whole ranking list into the item representations. PRM [33] and SetRank [32] adopt the self-attention mechanism to model the influence between any pair of items in the list. Yet the performance of list-level reranking algorithms is usually suboptimal when the recommendation page presented to the user is in a multi-list style.

Hao et al. [19] find the information from other lists on the page can improve the performance of reranking, and propose a deep and hierarchical attention network reranking (DHNAR) model. They aggregate the page-wise context into a static page representation vector, and apply an identical list-level reranking algorithm for all the lists with the page representation as shared side information. However, DHNAR fails to capture the dynamic page-wise interaction between items, and is insensitive to the different page formats or the commonalities and distinctions between lists. Our proposed PAR fully exploits the page-wise context and captures multifaceted fine-grained item influences across lists.

5.2 Multi-scenario Learning to Rank

Multi-scenario learning to rank aims to improve the overall performance in different scenarios, which can be generally classified into two categories: (1) multi-task learning (MTL) [6, 16, 23, 43], and (2) multi-agent reinforcement learning (MARL) [11, 20]. MTL-based methods formulate the ranking problems from different scenarios as different tasks, and devise a single model to solve the multiple tasks simultaneously. As for MARL-based methods, each scenario has a local ranking agent, and the agents are trained collaboratively to improve the overall performance. However, the agents are updated in an online setting with instant feedback, which is different from ours.

Page-level reranking intends to simultaneously rerank lists on the same page, which emphasizes on user behaviors when examining a recommendation page as a whole. How to model the page format and contexts is key to page-level reranking, which is ignored in multi-scenario learning to rank.

5.3 Whole-page Optimization

Whole-page optimization focuses on improving the display of the recommendation page, which generally falls into two categories. The first category [31, 35–37, 42] aims to find the optimal presentation style for each item on the page. Presentation style includes positions, image sizes, text fonts, etc. The problem is then formulated as a combinatorial optimization problem of determining positions and other presentation styles for each item, where graph matching [36, 37], bandit [35], or reinforcement learning (RL) [31, 37, 42] algorithms are proposed. This type of method, however, is designed for mapping one single initial list into a 2D geometrical layout, which is different from our work — the input of PAR is multiple initial lists of different themes. The second category [5, 9, 15, 27] is designed to select and rank the widgets (list of items) of the page. Given a set of widgets where the order of items for each widget is fixed, these methods try to select personalized themes of widgets to meet users’ needs. These models, though involving page-level information, are not the focus of our work. In this work, we jointly optimize the arrangement of items for multiple lists by considering the page-wise context.

6 CONCLUSION

In this work, we study the problem of page-level reranking, which requires a unified model to rerank multiple lists simultaneously on the same recommendation page. We conduct a data-driven study based on a real-world multi-list dataset, and propose a novel Page-level Attentional Reranking (PAR) model. We design a hierarchical dual-side attention module and a spatial-scaled attention network to learn the fine-grained spatial-aware item interactions across lists. Besides, we adopt the multi-gated mixture-of-experts module to capture the commonalities and distinctions of user behaviors among different lists. Extensive experiments show that PAR significantly outperforms the state-of-the-art baselines.

ACKNOWLEDGEMENT

The SJTU team is supported by Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and National Natural Science Foundation of China (62177033). The work is also sponsored by Huawei Innovation Research Program. We thank MindSpore [1] for the partial support of this work.
REFERENCES

[1] 2020. MindSpore. https://www.mindspore.cn/
[2] Qingyao Ai, Kepei Bi, Jiafeng Guo, and W Bruce Croft. 2018. Learning a deep listwise context model for ranking refinement. In The 41st international ACM SIGIR conference on research & development in information retrieval. 135–144.
[3] Qingyao Ai, Xuanhui Wang, Sebastian Brach, Nadav Golbandi, Michael Bender, and Marc Najork. 2019. Learning groupwise multivariate scoring functions using deep neural networks. In SIGIR. 85–92.
[4] Irwan Bello, Sayali Kulkarni, Sagar Jain, Craig Boutilier, Ed Chi, Elad Eban, Xiyang Luo, Alan Mackey, and Ofer Meshi. 2018. Seq2label: Re-ranking and slate optimization with rnmns. arXiv preprint arXiv:1810.02019 (2018).
[5] WaiL Bendada, Guillaume Salha, and Théo Bontempelli. 2020. Carousel personalization in music streaming apps with contextual bands. In Fourteenth ACM Conference on Recommender Systems. 420–425.
[6] Yuting Chen, Yanshi Wang, Yabo Ni, An-Xiang Zeng, and Lanfen Lin. 2020. Scenario-aware and Mutual-based approach for Multi-scenario Recommendation in E-Commerce. In 2020 International Conference on Data Mining Workshops (ICDMW). IEEE. 127–135.
[7] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In Proceedings of the 2008 international conference on web search and data mining. 87–94.
[8] Xinyi Dai, Jiangzhao Lin, Weichen Zhang, Shuai Li, Weiwen Liu, Ruiming Tang, Xiuqiang He, Jiaoye Hao, Jun Wang, and Yong Yu. 2021. An Adversarial Imitation Click Model for Information Retrieval. In Proceedings of the Web Conference 2021. 1809–1820.
[9] Weiting Ding, Dinesh Govindaraj, and SVN Vishwanathan. 2019. Whole page optimization with global constraints. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 3153–3161.
[10] Zhengqiao Du, Xiaowei Wang, Hongxia Yang, Jingen Zhou, and Jie Tang. 2019. Sequential Scenario-Specific Meta Learner for Online Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2895–2904.
[11] Jun Feng, Heng Li, Minlie Huang, Shichen Liu, Wenwu Ou, Zhongqiang Wang, and Xiaoyan Zhu. 2018. Learning to collaborate: Multi-scenario ranking via multi-agent reinforcement learning. In Proceedings of the 2018 World Wide Web Conference. 1939–1940.
[12] Yufei Feng, Yu Gong, Fei Sun, Junfeng Ge, and Wenwu Ou. 2021. Revisit recommender system in the permutation perspective. arXiv preprint arXiv:2102.12057 (2021).
[13] Yufei Feng, Binglin Hu, Yu Gong, Fei Sun, Qingwen Liu, and Wenwu Ou. 2021. GRN: Generative Rerank Network for Context-wise Recommendation. arXiv preprint arXiv:2104.00860 (2021).
[14] Lingyue Fu, Jiangzhuo Lin, Weiven Liu, Ruiming Tang, Weichen Zhang, Rui Zhang, and Yong Yu. 2022. An F-shape Click Model for Information Retrieval on Multi-block Mobile Pages. arXiv preprint arXiv:2206.08604 (2022).
[15] Carlos A Gomez-Uribe and Neil Hunt. 2015. The netflix recommender system: A Review. arXiv preprint arXiv:2202.06602 (2022).
[16] Chieb Luo, Hongliang Yu, Xin Yin, Krutika Shetty, Changhe He, Kathy Hu, Justin M Platz, Adam Ilardi, and Sriganesh Madhavan. 2021. Page-level Optimization of e-Commerce Item Recommendations. In Fifteenth ACM Conference on Recommender Systems. 495–504.
[17] Felicia Loescherbach, Kasper Welbers, Judith Moeller, Damian Trilling, and Wouter Van Atteveldt. 2021. Is This a Click towards Diversity? Explaining When and Why Users Make Diverse Choices. In 13th ACM Web Science Conference 2021. 282–290.
[18] Jiasen Liu, Jiawei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical Question-Image Co-Attention for Visual Question Answering. 289–297.
[19] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-task mixture-of-experts. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 1930–1939.
[20] Harrie Oosterhuis and Maarten de Rijke. 2018. Ranking for relevance and display positions in complex presentation layouts. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 845–854.
[21] Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2020. Setrank: Learning a permutation-invariant ranking model for information retrieval. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 499–508.
[22] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, et al. 2019. Personalized re-ranking for recommendation. In Proceedings of the 13th ACM conference on recommender systems. 3–11.
[23] Jiasui Qiu, Weian Zhang, Xin Wu, Jiawei Jin, Yuchen Fang, and Yong Yu. 2020. User behavior retrieval for click-through rate prediction. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2347–2356.
[24] Yingfei Wang, Hua Ouyang, Chu Wang, Jianhui Chen, Tetvan Asamov, and Yi Chang. 2017. Efficient ordered combinatorial semi-bands for whole-page recommendation. In Proceedings of the 2017 Conference on Artificial Intelligence.
[25] Yue Wang, Dawei Yin, Luo Jie, Pengyu Wang, Makoto Yamada, Yi Chang, and Qiaozhu Mei. 2016. Beyond ranking: Optimizing whole-page presentation. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. 101–112.
[26] Yue Wang, Dawei Yin, Luo Jie, Pengyu Wang, Makoto Yamada, Yi Chang, and Qiaozhu Mei. 2018. Optimizing whole-page presentation for web search. ACM Transactions on the Web (TWEB) 12, 3 (2018), 1–25.
[27] Chuan Wu, Fangzhao Wu, and Yongfeng Huang. 2021. DA-Transformer: Distance-aware Transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2059–2068.
[28] Yunju Xu, Weiven Liu, Jierong Zhu, Xilong Zhao, Xinyi Dai, Ruiming Tang, Weichen Zhang, Rui Zhang, and Yong Yu. 2022. Multi-Level Interaction Re-ranking with User Behavior History. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.
[29] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 1061–1067.
[30] Xiaodong He, Jianye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Optimization with Global Constraints. In The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 1259–1268.
[31] Jiangzhao Lin, Weiren Liu, Xinyi Dai, Weihan Zhang, Shuai Li, Ruiming Tang, Xiuqiang He, Jiaoye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Web Search. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 1259–1268.
[32] Weiren Liu, Yunjia Xi, Xinjia Qiu, Fei Sun, Bo Chen, Weihan Zhang, Rui Zhang, and Ruiming Tang. 2022. Neural Re-ranking in Multi-stage Recommender Systems: A Review. arXiv preprint arXiv:2202.06602 (2022).
[33] Yi Li, Jierong Zhu, Weiren Liu, Liangcai Su, Guohao Cai, Qi Zhang, Ruiming Tang, Xi Xiao, and Xiuqiang He. 2022. PEAR: Personalized Re-ranking with Contextualized Transformer for Recommendation. arXiv preprint arXiv:2203.12267 (2022).