

RIA: A Ranking-Infused Approach for Optimized listwise CTR Prediction

Guoxiao Zhang*
Tan Qu*
zhangguoxiao@meituan.com
qutan@meituan.com
Meituan
Beijing, China

Ao Li*
Meituan
Beijing, China
liao27@meituan.com

Donglin Ni
Beijing University of Posts and
Telecommunications
Beijing, China
nidl@bupt.edu.cn

Qianlong Xie
Meituan
Beijing, China
xieqianlong@meituan.com

Xingxing Wang
Meituan
Beijing, China
wangxingxing04@meituan.com

Abstract

Reranking improves recommendation quality by modeling item interactions. However, existing methods often decouple ranking and reranking, leading to weak listwise evaluation models that suffer from combinatorial sparsity and limited representational power under strict latency constraints. In this paper, we propose **RIA** (Ranking-Infused Architecture), a unified, end-to-end framework that seamlessly integrates pointwise and listwise evaluation. **RIA** introduces four key components: (1) **the User and Candidate Dual-Transformer (UCDT)** for fine-grained user-item-context modeling; (2) **the Context-aware User History and Target (CUHT)** module for position-sensitive preference learning; (3) **the Listwise Multi-HSTU (LMH)** module to capture hierarchical item dependencies; and (4) **the Embedding Cache (EC)** module to bridge efficiency and effectiveness during inference. By sharing representations across ranking and reranking, **RIA** enables rich contextual knowledge transfer while maintaining low latency. Extensive experiments show that **RIA** outperforms state-of-the-art models on both public and industrial datasets, achieving significant gains in AUC and LogLoss. Deployed in Meituan's advertising system, **RIA** yields a +1.69% improvement in Click-Through Rate (CTR) and a +4.54% increase in Cost Per Mille (CPM) in online A/B tests.

CCS Concepts

• Information systems → Information retrieval.

Keywords

CTR Prediction, Rerank, Recommendation Systems

*Both authors contributed equally to this research.

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 the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2018/06
<https://doi.org/XXXXXXX.XXXXXXX>

ACM Reference Format:

Guoxiao Zhang, Tan Qu, Ao Li, Donglin Ni, Qianlong Xie, and Xingxing Wang. 2018. RIA: A Ranking-Infused Approach for Optimized listwise CTR Prediction. In *Proceedings of Companion Proceedings of the ACM Web Conference 2026 (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

Reranking aims to optimize the final item list by modeling interactions among candidates, enabling listwise utility maximization. While early generator-only methods [1, 3, 6] suffer from the *evaluation-before-reranking* dilemma [6], recent evaluation-based approaches [2, 4, 7] adopt a two-stage paradigm: generating candidate lists followed by listwise scoring. However, these methods primarily focus on list generation, leaving the listwise evaluation model under-explored. More recently, the evaluation-only method YOLOR [5] bypasses list generation but suffers from two limitations: it implicitly assumes conditional independence between position and context, and its computational cost limits scalability to millions or more candidate lists.

A key challenge for listwise evaluation is *combinatorial sparsity*, the exponential decay in co-exposure frequency as list size increases. As illustrated in Figure 1, while individual items may be frequently exposed, their joint occurrences (e.g., triplets) are extremely rare, making it difficult to learn robust contextual interactions from data. Compounding this issue, those listwise evaluation model are architecturally decoupled from the pointwise evaluation models during the initial ranking stage. Designed under strict latency constraints¹, they tend to be simpler and less expressive, creating a representational gap that hinders knowledge transfer and limits modeling capacity.

To address these limitations, we propose **RIA** (Ranking-Infused Architecture), a unified evaluation-based framework that seamlessly integrates pointwise and listwise evaluation models into a single end-to-end pipeline. **RIA** consists of four components: (1) **the User and Candidate Dual-Transformer (UCDT)** for fine-grained candidate-context interaction modeling; (2) **the Context-aware**

¹The rerank stage processes fewer candidates than the rank stage, so the recommendation pipeline allocates less time consumption to it.

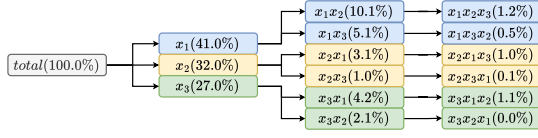


Figure 1: An illustration of *combinatorial sparsity*.

User History and Target (CUHT) module for position-aware preference learning; (3) the **Listwise Multi-HSTU (LMH)** module for hierarchical cross-item dependency modeling; (4) the **Embedding Cache (EC)** module to enhance inference efficiency.

Our contributions are threefold: (i) We present **RIA**, the first framework to unify pointwise and listwise evaluation under a single end-to-end paradigm; (ii) We design a powerful yet efficient listwise evaluation model with novel modules, which outperforms state-of-the-art models on both public and industrial datasets; (iii) We deploy **RIA** on Meituan’s advertising system, where online A/B tests show a +1.69% gain in CTR and +4.54% in CPM, validating its real-world effectiveness.

2 Methodology

Our framework, illustrated in Figure 2, seamlessly integrates pointwise and listwise evaluation through four synergistic modules: **UCDT**, **CUHT**, **LMH**, and **EC**. For each user u , given a candidate list $X = \{x_1, \dots, x_n\}$ from the ranking stage and a set of generated target candidate lists \mathcal{P} by generative models, where each target candidate list $P \in \mathcal{P}$ contains m items ($m \leq n$), the goal is to select the optimal list P^* that maximizes the expected listwise reward:

$$P^* = \arg \max_{P \in \mathcal{P}} R(u, P), \quad \text{where} \quad R(u, P) = \sum_{i=1}^m r(u, x_i). \quad (1)$$

Here, $r(u, x_i)$ denotes the reward for item x_i , and $R(u, P)$ represents the cumulative reward of the entire list P .

2.1 User and Candidate Dual-Transformer (UCDT)

The **UCDT** module captures fine-grained interactions between candidates and user context. Following the approach of [8], we merge user profile features, context features into the time series composed with user pointwise behaviors², which is called user-context features (defined as \mathbf{e}^u). Let $\mathbf{X} \in \mathbb{R}^{n \times D}$ and $\mathbf{E}^u \in \mathbb{R}^{T \times D}$ denote embeddings of candidate list and user-context features, where n is the number of candidate items, D is the dimension of the embedding layer and T is the length of the time series.

Both \mathbf{E}^u and \mathbf{X} are encoded using Hierarchical Sequential Transduction Units (HSTU) [8]:

$$\mathbf{X}' = \text{HSTU}(\mathbf{X}), \quad (2)$$

$$\mathbf{E}^{u'} = \text{HSTU}(\mathbf{E}^u). \quad (3)$$

We then apply a target attention mechanism [10] to model the interaction between each candidate item and the user-context features:

$$\mathbf{x}_i'' = \text{Attention}(\mathbf{x}_i', \{\mathbf{e}_j^{u'}\}_{j=1}^T), \quad i = 1, \dots, n. \quad (4)$$

²which ignore the contextual items within each session.

where $\mathbf{x}_i' \in \mathbb{R}^D$ is the i -th candidate item in \mathbf{X}' , $\mathbf{e}_j^{u'} \in \mathbb{R}^D$ is the j -th user-context feature in $\mathbf{E}^{u'}$.

The pointwise CTR prediction \hat{y}_i^p is:

$$\hat{y}_i^p = \sigma(\text{MLP}(\mathbf{x}_i'')), \quad i = 1, \dots, n. \quad (5)$$

where σ is the sigmoid function. The corresponding loss is calculated as binary cross-entropy loss defined as \mathcal{L}_1 .

2.2 Context-aware User History and Target (CUHT)

In user behavior modeling, **UCDT** treats each action in isolation and ignores the contextual items within the same session. To address this limitation, we propose **CUHT**, as illustrated in Figure 3, which consists of two core components: (1) *Page-level Inner Attention Unit* (PIAU) that models intra-session context via self-attention over user permutation-level historical behaviors³ and the target candidate list; and (2) *Position-aware Target Attention Unit* (PTAU) that explicitly models the interaction between the target candidate list and the permutation-level historical behavior at each session position.

2.2.1 PIAU. Using shared embedding layers from **UCDT**, we denote $\mathbf{E}_k \in \mathbb{R}^{m \times D'}$ and $\mathbf{E}_{L+1} \in \mathbb{R}^{m \times D'}$ as the k -th permutation embeddings of user permutation-level historical behaviors with length L and the embeddings of the target candidate list⁴, where D' is the sum of D and the position embedding size. PIAU applies a parameter-share self-attention layer to calculate the mutual influence of different items and output corresponding matrix \mathbf{H}_k , as follows:

$$\mathbf{H}_k = \text{selfAttention}(\mathbf{E}_k), \quad k = 1, \dots, L + 1. \quad (6)$$

2.2.2 PTAU. Specifically, for a given position o in the session, PTAU computes target attention between the o -th item in the target candidate list and all items in the permutation-level historical behaviors at position o :

$$\mathbf{w}_o = \text{Attention}(\mathbf{h}_{o,L+1}, \{\mathbf{h}_{o,k}\}_{k=1}^L), \quad o = 1, \dots, m. \quad (7)$$

where $\mathbf{h}_{o,k}$ represents the final representation at the o -th position in \mathbf{H}_k , and $\mathbf{h}_{o,L+1}$ represents the final representation in \mathbf{H}_{L+1} .

2.3 Listwise Multi-HSTU (LMH)

LMH models hierarchical list-level dependencies. It first transforms middle representations \mathbf{x}_o'' from Equation (4) via a MLP adaptor:

$$\mathbf{t}_o = \text{MLP}(\mathbf{x}_o''), \quad o = 1, \dots, m. \quad (8)$$

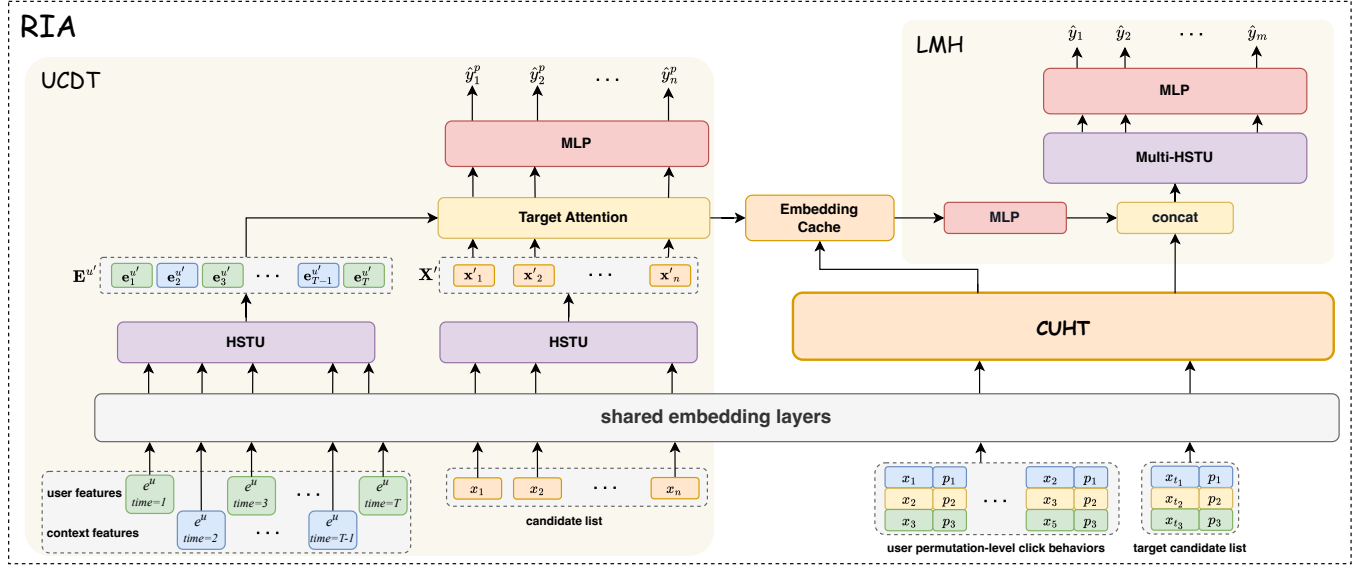
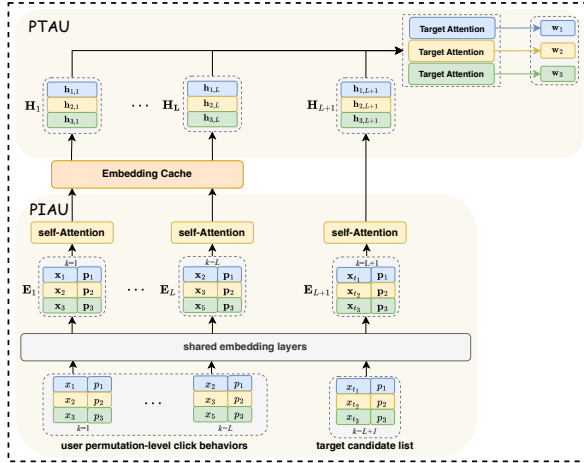
The target list representation is built by stacking HSTU layers (with layer num as I) over concatenated \mathbf{t}_o and \mathbf{w}_o :

$$\mathbf{M}_i = [\mathbf{m}_{i,1}, \dots, \mathbf{m}_{i,m}] = \begin{cases} \text{HSTU}([\mathbf{t}_1 \parallel \mathbf{w}_1, \dots, \mathbf{t}_m \parallel \mathbf{w}_m]), & i = 1, \\ \text{HSTU}(\mathbf{M}_{i-1}), & i = 2, \dots, I. \end{cases} \quad (9)$$

where \parallel denotes concatenation and $\mathbf{m}_{i,o}$ is the features of the o -th position in \mathbf{M}_i . Then, the listwise pCTR of the o -th item in the

³which incorporate session-level contextual information from past user interactions.

⁴The position embeddings are included.

Figure 2: Overview of our proposed method ($m = 3$).Figure 3: Architecture of CUHT ($m = 3$).

target list is predicted as follows:

$$\hat{y}_o = \sigma(\text{MLP}(\mathbf{m}_{l,o})), \quad o = 1, \dots, m. \quad (10)$$

Subsequently, the listwise loss is binary cross-entropy loss defined as \mathcal{L}_2 . Finally, the total loss is:

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2. \quad (11)$$

2.4 Embedding Cache (EC)

Due to strict time constraints, such a complex method cannot be applied directly to the reranking module. Therefore, we further design an EC module to enable efficient reuse. During ranking, We pre-compute the item-level representation \mathbf{x}_i' and the user permutation-level click behaviors representation \mathbf{H}_k through the

aforementioned UCDDT and PIAU in CUHT modules, respectively, as shown in Figure 2 and Figure 3.

3 EXPERIMENTS

3.1 Experimental Setup

3.1.1 Datasets. We evaluate our framework on both a public benchmark and a large-scale industrial dataset. The public **Avito** dataset contains user search and ad interaction logs. The industrial **Meituan** dataset is collected from Meituan’s advertising system, reflecting real-world user behavior in a large-scale local services ecosystem. Key statistics are summarized in Table 1.

Table 1: Dataset statistics.

Dataset	#Requests	#Users	#Items
Avito	53,562,269	1,324,103	23,562,269
Meituan	88,279,996	24,074,754	9,190,395

3.1.2 Baselines. We compare our model against representative pointwise and listwise CTR models:

- **PRM** [3]: Uses self-attention to model item dependencies in reranking.
- **OCPM** [4]: Listwise model with omnidirectional attention and context-aware prediction.
- **YOLOR** [5]: State-of-the-art listwise model with multi-scaling context information.
- **RIA_small**: Our model with 1 HSTU layer in the LMH module.
- **RIA_big**: Our model with 8 HSTU layers.

3.1.3 Evaluation Metrics. We report **AUC** and **LogLoss** for offline evaluation, following standard practice in CTR prediction.

3.2 Overall Performance

Table 2 presents the offline performance comparison. Our proposed **RIA** consistently outperforms all baselines on both datasets.

On the Avito dataset, **RIA_small** improves AUC by +0.40% over YOLOR, while **RIA_big** achieves a +0.85% gain. On the larger Meituan dataset, the improvements of AUC are even more pronounced: +0.31% (**RIA_small**) and +0.96% (**RIA_big**). These results validate the effectiveness of our unified architecture in capturing both fine-grained interactions and listwise dependencies.

Table 2: Offline performance comparison.

Model	Avito		Meituan	
	AUC	LogLoss	AUC	LogLoss
PRM	0.7131	0.0481	0.6541	0.1614
OCPM	0.7320	0.0471	0.6624	0.1596
YOLOR	0.7340	0.0470	0.6634	0.1595
RIA_small	0.7380	0.0468	0.6665	0.1592
RIA_big	0.7425	0.0456	0.6730	0.1483

3.3 Scaling Behavior of the LMH Module

We analyze the impact of architectural depth in the **LMH** module by varying the number of HSTU layers on Meituan dataset. As shown in Figure 4, AUC increases monotonically with depth on the Meituan dataset, rising from 0.6665 (1 layer) to 0.6730 (8 layers). This consistent improvement suggests a *scaling law* in listwise modeling.

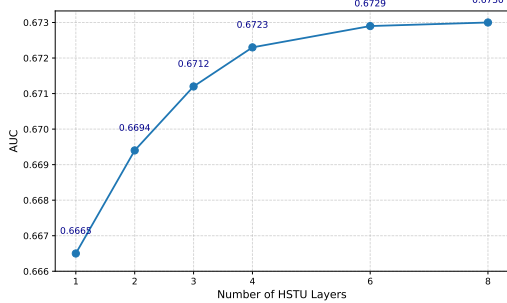


Figure 4: AUC vs. Number of HSTU layers.

3.4 Online Deployment and A/B Test Results

To enable real-time inference, we deploy a hybrid optimization strategy combining pre-computation and hierarchical caching, inspired by MSD [9]. Specifically, the **UCDT** module and **PIAU** component are computed in parallel with the ranking pipeline. Embeddings for top- n candidates and recent user behavior sequences are pre-computed and cached in Redis, significantly reducing online latency.

We conducted an A/B test on Meituan’s advertising system from September 6 to 14, 2025. As shown in Table 3, **RIA_small** achieves a

+1.69% increase in CTR and +4.54% in CPM over the baseline⁵ while keeping latency nearly unchanged (26.1 ms to 28.2 ms). **RIA_big** achieves even greater performance gains (+2.11% CTR, +5.83% CPM). These results demonstrate the practical effectiveness and business value of our framework.

Table 3: Online A/B testing results.

Method	CTR Gain	CPM Gain	Latency
Baseline	-	-	26.1 ms
RIA_small	+1.69%	+4.54%	28.2 ms
RIA_big	+2.11%	+5.83%	36.7 ms

4 Conclusion

In this paper, we propose **RIA**, a unified framework for end-to-end listwise CTR prediction that seamlessly integrates ranking and reranking. Extensive experiments show that **RIA** outperforms state-of-the-art baselines, achieving significant AUC gains on both public and industrial datasets. Our work bridges the gap between ranking and reranking through a single, trainable architecture, offering both performance gains and deployment efficiency. We hope this unified paradigm inspires future research on holistic, context-aware ranking systems in large-scale recommendation scenarios.

References

- [1] Qingyao Ai, Keping 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.
- [2] Yufei Feng, Yu Gong, Fei Sun, Junfeng Ge, and Wenwu Ou. 2021. Revisit recommender system in the permutation prospective. *arXiv preprint arXiv:2102.12057* (2021).
- [3] 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.
- [4] Xiaowen Shi, Fan Yang, Ze Wang, Xiaoxu Wu, Muzhi Guan, Guogang Liao, Wang Yongkang, Xingxing Wang, and Dong Wang. 2023. PIER: Permutation-Level Interest-Based End-to-End Re-ranking Framework in E-commerce. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4823–4831.
- [5] Shuli Wang, Yinqiu Huang, Changhao Li, Yuan Zhou, Yonggang Liu, Yongqiang Zhang, Yinhua Zhu, Haitao Wang, and Xingxing Wang. 2025. You Only Evaluate Once: A Tree-based Rerank Method at Meituan. *arXiv preprint arXiv:2508.14420* (2025).
- [6] Yunjia Xi, Weiwen Liu, Xinyi Dai, Ruiming Tang, Weinan Zhang, Qing Liu, Xiuqiang He, and Yong Yu. 2022. Context-aware Reranking with Utility Maximization for Recommendation. *arXiv:2110.09059 [cs.LG]* <https://arxiv.org/abs/2110.09059>
- [7] Hailan Yang, Zhenyu Qi, Shuchang Liu, Xiaoyu Yang, Xiaobei Wang, Xiang Li, Lantao Hu, Han Li, and Kun Gai. 2025. Comprehensive List Generation for Multi-Generator Reranking. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2298–2308.
- [8] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. 2024. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152* (2024).
- [9] Guoxiao Zhang, Yi Wei, Yadong Zhang, Huajian Feng, and Qiang Liu. 2025. Balancing Efficiency and Effectiveness: An LLM-Infused Approach for Optimized CTR Prediction. In *Companion Proceedings of the ACM on Web Conference 2025*. 596–600.
- [10] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

⁵Since YOLOR is not suited to our scenario, we adopt OCPM as the baseline.