

HHFT: Hierarchical Heterogeneous Feature Transformer for Recommendation Systems

Liren Yu
Taobao & Tmall Group of Alibaba
Hangzhou, China
yuliren.ylr@taobao.com

Wenming Zhang
Taobao & Tmall Group of Alibaba
Hangzhou, China
zhangwenming.zwm@taobao.com

Silu Zhou
Taobao & Tmall Group of Alibaba
Hangzhou, China
silu.zsl@taobao.com

Zhixuan Zhang
Taobao & Tmall Group of Alibaba
Hangzhou, China
zhixuan.zzx@taobao.com

Dan Ou
Taobao & Tmall Group of Alibaba
Hangzhou, China
oudan.od@taobao.com

Abstract

We propose HHFT (Hierarchical Heterogeneous Feature Transformer), a Transformer-based architecture tailored for industrial CTR prediction. HHFT addresses the limitations of DNN through three key designs: (1) Semantic Feature Partitioning: Grouping heterogeneous features (e.g. user profile, item information, behaviour sequence) into semantically coherent blocks to preserve domain-specific information; (2) Heterogeneous Transformer Encoder: Adopting block-specific QKV projections and FFNs to avoid semantic confusion between distinct feature types; (3) Hiformer Layer: Capturing high-order interactions across features. Our findings reveal that Transformers significantly outperform DNN baselines, achieving a +0.4% improvement in CTR AUC at scale. We have successfully deployed the model on Taobao's production platform, observing a significant uplift in key business metrics, including a +0.6% increase in Gross Merchandise Value (GMV).

CCS Concepts

• **Information systems** → *Retrieval models and ranking.*

Keywords

Recommendation System, Ranking Model, Scaling Laws,

ACM Reference Format:

Liren Yu, Wenming Zhang, Silu Zhou, Zhixuan Zhang, and Dan Ou. 2018. HHFT: Hierarchical Heterogeneous Feature Transformer for Recommendation Systems. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

Recommendation systems serve as the cornerstone of modern digital ecosystems, particularly in e-commerce platforms, where they directly influence user engagement and business revenue through

personalized content delivery. Over the past decade, recommendation technologies have evolved from traditional collaborative filtering to deep learning-based approaches, with DNNs becoming the de facto standard for CTR prediction tasks [3, 4, 6, 13, 15]. However, despite their universal approximation capabilities [9], DNN-based methods still struggle to explicitly model high-order feature interactions in sparse, high-dimensional user-item-context data. Their feedforward architectures fail to preserve semantically meaningful cross-feature relationships [1, 13].

This limitation is especially evident in e-commerce search scenarios, where ranking performance depends on the nuanced interplay of user intent (e.g., search queries), item attributes (e.g., price, category), and contextual factors (e.g., time, device). Classic DNN-based models such as Wide&Deep[3], DeepFM[6] and DCNv2[13] attempt to address this issue through manual feature engineering or implicit interaction modeling, but they still struggle with scalable high-order interaction learning and semantic preservation for heterogeneous features.

The Transformer architecture, originally proposed in natural language processing (NLP) [12], has introduced a paradigm shift in sequence modeling through its self-attention mechanism, which dynamically captures pairwise dependencies between input elements. This capability enables adaptive and interpretable feature interaction learning [12], making it particularly suitable for recommendation tasks. Furthermore, its predictable scaling behavior [8] offers a clear path for performance gains in data-rich industrial settings. While recent studies have explored Transformers for recommendations, critical gaps remain. For example, SASRec [7] and LONGER [2] leverage the Transformer structure to capture long-term dependencies in user behavior sequences. However, these approaches primarily focus on sequential data and overlook broader heterogeneous feature interactions. Hiformer [5] explicitly model heterogeneous feature interactions but lack a rigorous empirical analysis of scaling laws. Wukong [14] and Rankmixer [16] explore the potential of scaling laws for explicit high-order feature interaction in recommendations, but rely on non-Transformer elements (e.g., Factorization Machine blocks in Wukong or MLP-mixer [11] in Rankmixer), limiting their ability to fully exploit Transformer's semantic-aware capabilities.

To bridge these gaps, this paper proposes Hierarchical Heterogeneous Feature Transformer (HHFT) tailored for CTR prediction

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, 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.

Conference acronym 'XX, 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>

in ranking systems. HHFT addresses two core limitations of DNN-based rankers: (1) **Explicit High-Order Interaction Modeling**: By integrating semantic feature partitioning and heterogeneous attention mechanisms, HHFT directly models high-order interactions through learnable affinity matrices, overcoming the implicit and compressed feature fusion of DNNs. (2) **Scalable Representation Learning**: We validate that HHFT adheres to scaling laws, enabling predictable performance gains with increased model capacity.

Our key contributions are threefold:

- (1) **Architectural Innovation**: We propose a hierarchical heterogeneous feature processing framework that partitions features by semantics and maintains domain-specific parameters, preserving the uniqueness of heterogeneous features while enabling cross-domain interactions.
- (2) **Empirical Validation of Scaling Laws**: We establish and validate predictable scaling relationships between model size and CTR prediction performance, providing a quantitative guide for model scaling in industrial settings.
- (3) **Industrial Deployment and Business Impact**: We successfully deploy HHFT on Taobao's production search platform, achieving statistically significant improvements in both model performance (AUC) and business metrics (GMV, CTR), demonstrating its practical application value.

2 Related Work

2.1 DNN-Based CTR Models

DNN-based methods dominate CTR prediction by modeling non-linear interactions. Wide&Deep [3] combines a wide linear layer (memorization) with a deep network (generalization). DeepFM [6] integrates Factorization Machines (FMs) into DNNs to model low-order interactions explicitly. DCNv2 [13] uses cross layers to learn high-order interactions via feature cross operations. These models, however, rely on implicit interaction learning and struggle with heterogeneous features.

2.2 Transformer-Based Recommender Systems

Transformers have gained traction for their explicit attention mechanisms. SASRec [7] and LONGER [2] apply Transformer encoders to user behavior sequences, capturing long-term dependencies. AutoInt [10] replaces DNN layers with self-attention to model feature interactions, but uses shared parameters for all features. Hiformer [5] proposes heterogeneous attention for feature interactions, but lacks scaling law analysis.

2.3 Scaling Laws in Recommendation

Scaling laws, where performance improves predictably with model size, are critical for industrial platforms. Recent works like Wukong [14] RankMixer [16] explore scaling laws for recommendation, but Wukong relies on FM blocks and RankMixer designs token-mixing from mlpmixer [11] (not Transformers).

Scaling laws—where model performance improves in a predictable manner with increased model size—are of critical importance for industrial recommendation platforms. Recent works focusing on scaling laws for recommendation tasks include Wukong [14] and RankMixer [16]. However, Wukong relies on Factorization Machine

(FM) blocks for feature interaction modeling, and RankMixer designs its token-mixing mechanism based on MLP-mixer [11] instead of adopting Transformer-based attention and further ignores heterogeneous feature partitioning. These limitations mean that neither one of the solutions fully utilizes the strengths of the Transformer to capture complex, semantically aware feature interactions—a key gap addressed by our proposed HHFT.

This paper is structured as follows: Section 2 details the architectural of Transformer-based CTR ranking model; Section 3 presents scaling experiments and online results; Section 5 explores implications and future work.

3 Methodology

3.1 Overview

Our proposed **Hierarchical Heterogeneous Feature Transformer (HHFT)** replaces traditional DNN backbones with a Transformer-based structure optimized for recommendation scenarios. As shown in Figure 1, HHFT processes input features through five stages:

- (1) **Feature Partitioning**: Group raw features into semantically coherent blocks to preserve domain-specific information.
- (2) **Heterogeneous Feature Tokenization**: Convert each feature block into unified dimension tokens through embedding and projection, enabling inter-block computation.
- (3) **Heterogeneous Transformer Encoder**: Model interactions between feature tokens using Transformer with domain-specific parameters.
- (4) **Hiformer Layer**[5]: Enhance high-order interaction across all features through a sophisticated attention mechanism.
- (5) **Prediction Head**: Generate final CTR/CVR predictions using an MLP.

3.2 Feature Tokenization

The features of E-commerce recommendation/search system exhibit high heterogeneity, including categorical features (e.g., user ID, item category), continuous features (e.g., item price, user purchase frequency) and sequential features (e.g., user recent behavior). Directly concatenating these features leads to semantic information loss. Thus, we first partition the input features into K disjoint blocks based on semantic categories:

$$\mathcal{B} = \{ \underbrace{\mathbf{U}}_{\text{user features}}, \underbrace{\mathbf{I}}_{\text{item features}}, \underbrace{\mathbf{Q}}_{\text{query features}}, \underbrace{\mathbf{S}}_{\text{behavior sequence}}, \dots \} \quad (1)$$

Each block $\mathbf{B}_k \in \mathbb{R}^{d_k}$ has a different dimension. Through embedding layers:

$$\mathbf{E}_k = \text{Embed}_k(\mathbf{B}_k) \in \mathbb{R}^{e_k} \quad (2)$$

where e_k is the size of the block-specific embedding.

To enable cross-block attention, we project all embedding vectors to a unified dimension d using block-specific linear layers:

$$\mathbf{H}_k^{(0)} = \mathbf{W}_k^{\text{proj}} \mathbf{E}_k + \mathbf{b}_k^{\text{proj}}, \quad \mathbf{W}_k^{\text{proj}} \in \mathbb{R}^d \quad (3)$$

yields an aligned token matrix $\mathbf{H}^{(0)} = [\mathbf{H}_U, \mathbf{H}_I, \mathbf{H}_Q, \mathbf{H}_S, \dots]^T \in \mathbb{R}^{K \times d}$.

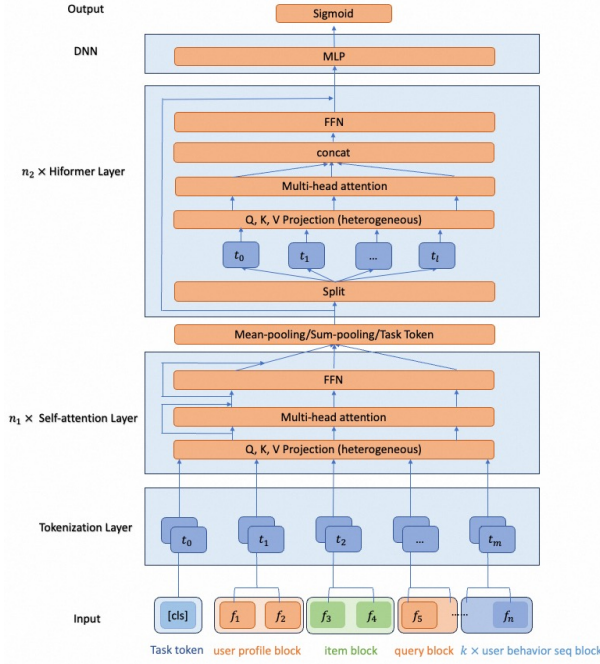


Figure 1: HHFT Architecture.

3.3 Heterogeneous Transformer Encoder

Traditional Transformer encoders adopt shared parameters for all input tokens, which fails to preserve the uniqueness of heterogeneous features (e.g., categorical user IDs, continuous item prices, and sequential behavior features) in e-commerce recommendation/search scenarios. This deficiency often leads to semantic information confusion and suboptimal interaction modeling. To address this issue, the Heterogeneous Transformer Encoder proposed in this paper maintains independent parameter sets for Query (Q), Key (K), Value (V) projection and Feed-Forward Network (FFN) for each feature block (user/item/query/behavior sequence...), while reusing the core attention computation logic to ensure both feature specificity and computational efficiency. The detailed implementation is as follows:

- (1) **Block-Specific QKV Projection** For k -th feature block at the l -th encoder layer, the token representation ($H_k^{(l)} \in \mathbb{R}^d$) is mapped to block-specific Q, K, V vectors using independent projection matrices. The calculation formula is: $Q_k = W_{Q,k} \cdot H_k^{(l)}$, $K_k = W_{K,k} \cdot H_k^{(l)}$, $V_k = W_{V,k} \cdot H_k^{(l)}$, where $W_{Q,k}, W_{K,k}, W_{V,k} \in \mathbb{R}^{d \times d}$ are block-specific weight. This design ensures that the attention weight calculation adapts to the semantic characteristics of each feature domain.
- (2) **Multi-Head Self-Attention** To capture high-order feature interactions, the Q, K, V vectors of each block are processed by standard multi-head self-attention.
- (3) **Block-Specific FFN** After attention computation, each block's token representation is input to a block-specific FFN to enhance nonlinear feature transformation capability. The FFN

adopts a two-layer fully connected structure with ReLU activation, and the formula is: $+FFN_k(x) = \text{ReLU}(x \cdot W_{1,k} + b_{1,k}) \cdot W_{2,k} + b_{2,k}$ where $W_{1,k} \in \mathbb{R}^{d \times d}$, $b_{1,k} \in \mathbb{R}^{4d}$, $W_{2,k} \in \mathbb{R}^{d \times d}$, $b_{2,k} \in \mathbb{R}^d$ are block-specific parameters. The hidden layer dimension is set to d to balance expressive power and computational cost.

3.4 Hiformer Layer

Building on the heterogeneous Transformer encoder, the Hiformer layer[5] introduces composite projections to model interactions beyond pairwise dependencies. We use a global composite projection:

$$[\hat{K}_1^h, \dots, \hat{K}_k^h] = \text{concat}([H_1^h, \dots, H_k^h])\hat{W}^h \quad (4)$$

where $\hat{W}^h \in \mathbb{R}^{kd \times kd_h}$. This design enables Hiformer to learn more comprehensive hierarchical interactions across features.

3.5 Prediction Head

After n_1 transformer layers and n_2 hiformer layers, we concatenate all tokens and generate the final CTR/CVR prediction via a MLP.

4 Experiment

To comprehensively validate the effectiveness, component contributions, scaling properties, and industrial applicability of HHFT, we conduct experiments on Taobao's real-world e-commerce data.

4.1 Experiment Settings

4.1.1 Datasets and Environment. To evaluate HHFT, we conducted both offline and online experiments on Taobao's e-commerce dataset, comprising billions of user-item interactions.

4.1.2 Evaluation Metrics. To evaluate model performance, we use AUC (Area Under the Curve) as the primary performance metrics. An AUC increase of 0.001 can be regarded as a confidently significant improvement. We also report model size (Params, in millions) and computational complexity (TFLOPs) to evaluate efficiency-scalability tradeoffs. For online experiments, we use GMV as the business-centric metric to reflect practical impact.

4.1.3 Baselines. We compare against the following widely recognized SOTA baselines: **DLRM-MLP**: which is the vanilla MLP for feature crossing as the experiment baseline; **DCNv2**[13] the sota of feature cross model. **AutoInt**[10], **Hiformer**[5] investigate transform-like architecture for ranking model. **Wukong**[14] and **Rankmixer**[16] investigate the scaling law for recommendation system.

4.2 Comparison with SOTA methods

The main results are summarized in Table 1. HHFT outperforms all SOTA baselines in both predictive accuracy (AUC) and scalability, aligning with our design goals. We can also observe that Transformer-based models (AutoInt, Hiformer, HHFT) outperform DNN/FM-based models (DCNv2, Wukong), confirming the value of explicit attention for feature interactions. Furthermore, HHFT's AUC gain (vs. DLRM-MLP) exceeds Hiformer and Wukong, attributed to its hierarchical heterogeneous design and scaling optimization.

Table 1: Performance comparison of recommendation models (best values in bold)

Model	AUC	Params(M)	TFLOPs
DLRM-MLP (base)	0.78157	15	0.42
DCNv2	0.78252	24	0.65
AutoInt	0.78663	150	1.19
HiFormer	0.78641	170	1.98
Wukong	0.78355	32	0.94
Rankmixer	0.78586	140	1.93
HHFT	0.78909	300	1.22

4.3 Ablation Study

To quantify the incremental contribution of each key component in HHFT, we conduct ablation experiments under a cold-start training scenario (with 7 consecutive days of cold-start training, where all dense layer weights initialized randomly). Table 2 presents the AUC gains relative to the DLRM-MLP baseline, with key insights as follows:

- (1) Transformer Architecture: Replacing DNN-MLP with a basic Transformer encoder already yields significant gains, confirming that self-attention outperforms MLP in modeling feature interactions.
- (2) Heterogeneous Parameterization Introducing token-specific QKV projections and FFNs further improves performance, validating that avoiding semantic confusion between heterogeneous features is critical for interaction modeling.
- (3) HiFormer Layer: Adding HiFormer layers enhances high-order interactions, demonstrating the value of moving beyond pairwise dependencies.
- (4) Weight Initialization Optimization: Tuning parameter initialization and hyperparameters (e.g., layer normalization placement, learning rate scheduling) delivers the large gain, highlighting the importance of industrial-friendly training optimizations for Transformer-based models.
- (5) Scaling Up: Scaling up model size improves performance, consistent with scaling laws.

The combined effect of these components results in a total +0.0117 AUC gain, confirming that each design choice in HHFT contributes meaningfully to its performance advantage.

Table 2: Ablation on components of HHFT.

Setting	auc gain vs MLP
MLP → Transformer	+0.0035
Heterogeneous Transformer	+0.0018
HiFormer Layer	+0.0011
Weight Initialization	+0.0040
Scaling up	+0.0034

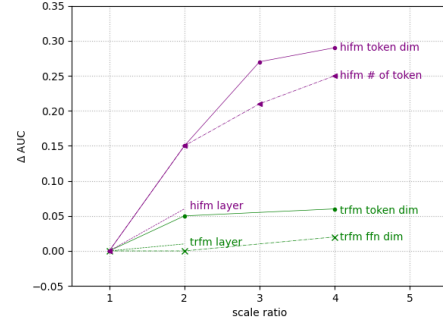
4.4 Scaling Laws Validation

A key advantage of Transformer-based models is compliance with scaling laws—performance improves predictably with model size. We define the "base model" of HHFT with the following dense parameters (excluding embeddings, which are fixed):

- Heterogeneous Transformer: layer ($n_1 = 1$), token dimension ($d_{\text{trfm}} = 1648$), FFN dimension ($d_{\text{ffn}} = 1648$)
- HiFormer Layer: layer ($n_2 = 1$), token dimension ($d_{\text{hifm}} = 256$), number of tokens ($n_h = 8$)

Then, we scale each parameter independently by a factor (keeping other parameters fixed), and measure AUC gain in Figure 2. Two critical conclusions emerge for industrial model scaling:

- (1) Width > Depth: Scaling the width of the model (e.g., Transformer/HiFormer token dimension) yields higher AUC gains than scaling depth (number of Transformer/HiFormer layers). This provides a practical guide for resource-efficient model expansion.
- (2) High-Order > Low-Order Scaling: Scaling parameters related to high-order interactions (e.g., HiFormer token count and token dimension) is more effective than scaling low-order components (e.g., Transformer layers). This confirms that HHFT's focus on hierarchical high-order interactions is not only architecturally innovative but also scaling-efficient.

**Figure 2: AUC gain vs Dense Parameters Scale Ratio**

4.5 Online Performance

To validate HHFT's industrial practicality, we conducted a 30-day A/B test on Taobao's search platform, allocating 1% of total traffic to the HHFT group (with DNN as the control baseline). Results show HHFT outperforms the DNN baseline by +0.4% in CTR AUC—an indicator reflecting accurate user click intent distinction—and drives a +0.6% increase in GMV. For Taobao's massive user base and transaction volume, these gains translate to substantial incremental revenue, fully confirming the model's real-world efficacy.

5 Conclusion and Future Work

This paper proposes HHFT, a hierarchical heterogeneous Transformer for e-commerce CTR prediction. By combining semantic feature partitioning, domain-specific parameters Transformer, and HiFormer layers, HHFT outperforms SOTA models and delivers significant online business gains. We also validate HHFT's compliance with scaling laws, providing a reliable path for industrial ranking model scaling.

Future work will focus on extending HHFT to joint ranking for search, recommendation, and advertising—sharing cross-domain features to further improve business value.

References

- [1] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. 2018. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 46–54.
- [2] Zheng Chai, Qin Ren, Xijun Xiao, Huizhi Yang, Bo Han, Sijun Zhang, Di Chen, Hui Lu, Wenlin Zhao, Lele Yu, et al. 2025. Longer: Scaling up long sequence modeling in industrial recommenders. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*. 247–256.
- [3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.
- [4] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.
- [5] Huan Gui, Ruoxi Wang, Ke Yin, Long Jin, Maciej Kula, Taibai Xu, Lichan Hong, and Ed H Chi. 2023. Hiformer: Heterogeneous feature interactions learning with transformers for recommender systems. *arXiv preprint arXiv:2311.05884* (2023).
- [6] 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).
- [7] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
- [8] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* (2020).
- [9] Hrushikesh N Mhaskar. 1996. Neural networks for optimal approximation of smooth and analytic functions. *Neural computation* 8, 1 (1996), 164–177.
- [10] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1161–1170.
- [11] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems* 34 (2021), 24261–24272.
- [12] 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).
- [13] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*. 1785–1797.
- [14] Buyun Zhang, Liang Luo, Yuxin Chen, Jade Nie, Xi Liu, Daifeng Guo, Yanli Zhao, Shen Li, Yuchen Hao, Yantao Yao, et al. 2024. Wukong: Towards a scaling law for large-scale recommendation. *arXiv preprint arXiv:2403.02545* (2024).
- [15] 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.
- [16] Jie Zhu, Zhifang Fan, Xiaoxie Zhu, Yuchen Jiang, Hangyu Wang, Xintian Han, Haoran Ding, Xinmin Wang, Wenlin Zhao, Zhen Gong, et al. 2025. RankMixer: Scaling Up Ranking Models in Industrial Recommenders. *arXiv preprint arXiv:2507.15551* (2025).