FINAL: Factorized Interaction Layer for CTR Prediction

Jieming Zhu*
Huawei Noah's Ark Lab, China

Qinglin Jia*
Huawei Noah's Ark Lab, China

Guohao Cai
Huawei Noah's Ark Lab, China

Quanyu Dai
Huawei Noah's Ark Lab, China

Jingjie Li
Huawei Noah's Ark Lab, China

Zhenhua Dong
Huawei Noah's Ark Lab, China

Ruiming Tang
Huawei Noah's Ark Lab, China

Rui Zhang
www.ruizhang.info

ABSTRACT

Multi-layer perceptron (MLP) serves as a core component in many deep models for click-through rate (CTR) prediction. However, vanilla MLP networks are inefficient in learning multiplicative feature interactions, making feature interaction learning an essential topic for CTR prediction. Existing feature interaction networks are effective in complementing the learning of MLPs, but they often fall short of the performance of MLPs when applied alone. Thus, their integration with MLP networks is necessary to achieve improved performance. This situation motivates us to explore a better alternative to the MLP backbone that could potentially replace MLPs. Inspired by factorization machines, in this paper, we propose FINAL, a factorized interaction layer that extends the widely-used linear layer and is capable of learning 2nd-order feature interactions. Similar to MLPs, multiple FINAL layers can be stacked into a FINAL block, yielding feature interactions with an exponential degree growth. We unify feature interactions and MLPs into a single FINAL block and empirically show its effectiveness as a replacement for the MLP block. Furthermore, we explore the ensemble of two FINAL blocks as an enhanced two-stream CTR model, setting a new state-of-the-art on open benchmark datasets. FINAL can be easily adopted as a building block and has achieved business metric gains in multiple applications at Huawei. Our source code will be made available at MindSpore/models and FuxiCTR/model_zoo.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender systems; CTR prediction; factorized interaction

ACM Reference Format:
Jieming Zhu, Qinglin Jia, Guohao Cai, Quanyu Dai, Jingjie Li, Zhenhua Dong, Ruiming Tang, Rui Zhang. 2023. FINAL: Factorized Interaction Layer for CTR Prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3539618.3591988

* Equal contribution. Corresponding to Jieming Zhu <jiemingzhu@ieee.org>.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591988

1 INTRODUCTION

Click-through rate (CTR) prediction is an important task in online advertising and recommendation, which aims to estimate the probability that a user will click on a certain item. Likewise, CTR prediction models can be applied to predicting users' like, favorite, purchase, or download actions. These tasks are usually formulated as a binary classification problem, which incorporates rich but heterogeneous features such as user profiles, item attributes, and session contexts [3]. As such, feature interaction learning becomes an important research topic for CTR prediction. Existing methods usually follow two directions to model feature interactions [10, 19]. The first one is using multi-layer perceptrons (MLP) to implicitly learn the hidden relations among features [3, 6, 20]. Although it is proved that MLPs can theoretically approximate any bounded continuous functions, in practice they are weak in modeling the combinatorial feature interactions given a limited network size [1, 21]. The second one is using multiplicative operations between features to explicitly model their interactions. For example, DCN [24], FM, xDeepFM [13], the feature combination degree in these methods is usually linearly proportional to the number of stacked layers, thereby a considerably deep architecture is required to comprehensively cover useful feature combinations [12, 15, 30]. However, it is difficult to optimize very deep models due to many extensively documented issues, such as gradient explosion/vanishment [11] and rank collapse [7, 8]. Thus, it is quite challenging for existing methods to effectively model high-order feature interactions.

In this paper, we propose a **F**actor**i**zed **In**tera**c**tion **L**ayer (FINAL) to explicitly learn multiplicative feature interactions, which can achieve very high combination orders without cumbersome layer stacking (Fig. 1). Motivated by the fast exponentiation algorithm, *FINAL* uses a hierarchical way to raise the feature interaction degree at an exponential speed. In each hierarchy, the input representations are multiplied with the representations output by a series of successive non-linear layers so that the feature interaction degree is progressively increased. By processing the feature representations with multiple hierarchies, the order of feature interactions is further exponentially increased. Based on the proposed *FINAL* module, we design a unified framework that incorporates multiple *FINAL* blocks to learn feature interactions in different views [2, 14], and we conduct self-distillation [9, 31] by using their prediction as common teachers to exchange the complementary knowledge they encode.

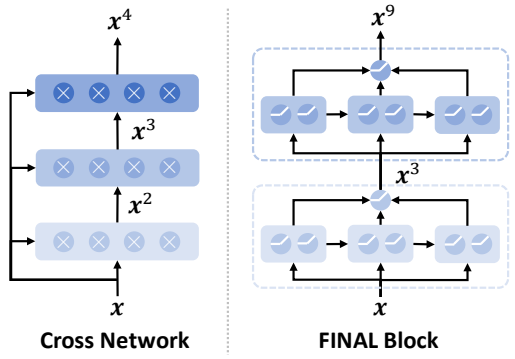

Figure 1: Comparison of the highest polynomial degree (in terms of input x) with the growth of staking layer depth.

We conduct extensive experiments on four public datasets and the results validate the superiority of FINAL. It also achieves notable success in online experiments in multiple commercial scenarios held by our enterprise. FINAL offers a new and simple option for composing CTR prediction models, which has the potential to empower various recommendation scenarios.

2 FINAL MODEL

2.1 Overall Framework

Our *FINAL* approach offers a universal framework for modeling feature interactions in CTR prediction. The overall framework is shown in Fig. 2. It is mainly composed of multiple *FINAL* blocks in parallel[1], which aims to model different feature interaction patterns. Each FINAL block contains several factorized interaction layers, where the maximum degree of feature interactions exponentially grows with the model depth. The prediction scores from different blocks are combined into a unified one as the final prediction, which is also served as a virtual teacher to self-teach these blocks to exchange and fuse their hidden knowledge. In this way, sophisticated feature interactions can be effectively and comprehensively captured.

2.2 FINAL Block

The basic unit of our approach is *FINAL* block. It receives a flattened feature vector $\mathbf{x}$ as the input, which may include various fields such as one-hot features, embedded categorical features, and numerical features. Due to the variety, complexity, and heterogeneity of crafted features in industrial scenarios, the interactions among features can often be complicated and implicit. Thus, modeling high-order interactions is critical for effective feature exploitation. In practice, how to achieve sufficiently high interaction orders with minimal model depths matters to both performance and efficiency. Motivated by the idea of fast exponentiation algorithms, we devise a hierarchical feature interaction mechanism to achieve exponential degree growth. In each hierarchy, a factorized interaction layer is used to raise the feature interaction degrees with several multiplicative operations. Denote the input of the l-th factorized interaction layer by $\mathbf{x}_{l-1}$. It is transformed using the following formulas:

$$\begin{aligned}
\mathbf{h}_{l,1} &= \mathbf{W}_{l,1}\mathbf{x}_{l-1} + \mathbf{b}_{l,1}, \\
\mathbf{h}_{l,2} &= \mathbf{h}_{l,1} \odot \sigma(\mathbf{W}_{l,2}\mathbf{x}_{l-1} + \mathbf{b}_{l,2}), \\
&\dots \\
\mathbf{h}_{l,N} &= \mathbf{h}_{l,N-1} \odot \sigma(\mathbf{W}_{l,N}\mathbf{x}_{l-1} + \mathbf{b}_{l,N}), \\
\mathbf{x}_l &= \sum_{i=1}^{N} (\mathbf{h}_{l,i}),
\end{aligned} \tag{1}$$

where $\mathbf{x}_l$ is the layer output, $\mathbf{W}_{l,i}$ and $\mathbf{b}_{l,i}$ are the weight and bias parameters associated the i-th operation, N is the number of multiplicative interaction steps, and σ is the activation function. Intuitively, the feature interaction degree is proportional to the number of multiplicative operations. By aggregating the intermediate results in each step, the output of each layer can include multi-granularity feature interactions. In a *FINAL* block, we stack multiple factorized interaction layers so that the initial feature interaction degree of each layer has been exponentially amplified by its precursors. In this way, the highest polynomial degree (in terms of elements in $\mathbf{x}$) for a K-layer final block with N multiplicative operations is N^K.

2.3 Cross-block Knowledge Transfer

In our approach, we prefer to use multiple *FINAL* blocks to learn feature interactions from different views. We first use different linear projection layers to convert the hidden representations into output logits. These logits are aggregated by average into a unified one, which is further normalized by the sigmoid function for model training (denoted as $\hat{y}$). We use the binary cross-entropy loss to calculate the CTR prediction loss as follows:

$$\mathcal{L}_c = -\frac{1}{S}\sum_{i=1}^{S}\left[y_i \log\left(\hat{y}_i\right) + \left(1-y_i\right)\log\left(1-\hat{y}_i\right)\right], \tag{2}$$

where y_i and $\hat{y}_i$ are the label and predicted score of the i-th sample, and S is the training data size. To facilitate knowledge sharing among different *FINAL* blocks, we perform self-knowledge distillation to empower them with inter-block knowledge. Concretely, we use the aggregated score $\hat{y}$ as the teacher, and encourage each block to learn from this synthesized prediction. Taking a dual-block network as an example (Fig. 2), we denote the normalized prediction scores of the two blocks by $\hat{y}_a$ and $\hat{y}_b$. Their corresponding knowledge distillation losses are as follows:

$$\begin{aligned}
\mathcal{L}_d &= -\frac{1}{S}\sum_{i=1}^{S}\left[\hat{y}_i \log\left(\hat{y}_{a,i}\right) + \left(1-\hat{y}_i\right)\log\left(1-\hat{y}_{a,i}\right)\right], \\
\mathcal{L}'_d &= -\frac{1}{S}\sum_{i=1}^{S}\left[\hat{y}_i \log\left(\hat{y}_{b,i}\right) + \left(1-\hat{y}_i\right)\log\left(1-\hat{y}_{b,i}\right)\right],
\end{aligned} \tag{3}$$

where $\hat{y}_{a,i}$ and $\hat{y}_{b,i}$ are the block-specific predictions on the i-th sample. We optimize the model with the task loss and knowledge transfer regularizations, and the overall loss function for model training is $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_d + \mathcal{L}'_d$. In this way, each block is aware of both the task supervision signals and cross-block knowledge, hence complicated feature interactions can be better covered.

[1]We use two FINAL blocks in our approach to trade-off performance and efficiency.

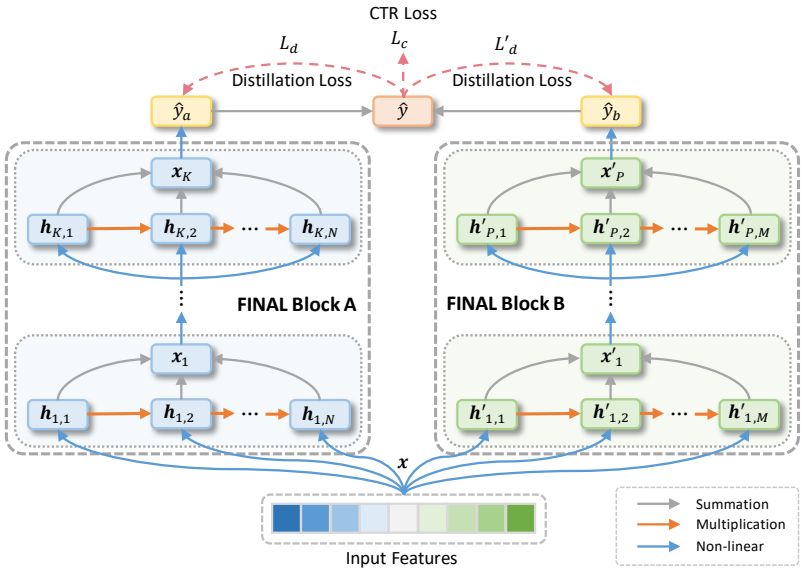

Figure 2: FINAL Blocks for CTR prediction.

2.4 Discussion

Finally, we present some brief discussions on model complexity and compatibility. Assuming the number of parallel blocks and hidden dimensions are constants, then the theoretical complexity of our framework is $O(NK)$. To achieve the same feature interaction degree, the required computational complexity of vanilla layer-stacking methods is usually $O(N^K)$. Thus, our approach has comparable efficiency with existing methods when N and K are small, and may have a substantial efficiency advantage when modeling extremely high-order feature interactions. Moreover, *FINAL* block is a plug-and-play module that can be directly inserted into existing architectures or replace their MLP-based feature interaction modules. Thus, *FINAL* is compatible with various CTR prediction methods and can empower them with light effort.

3 EXPERIMENTS

3.1 Experimental Settings

3.1.1 Datasets and Evaluation Metrics. We use four open benchmark datasets for experimental evaluation, including Criteo, Avazu, MovieLens, and Frappe. For fair comparison, we reuse the preprocessed datasets released by [4] and follow the same splitting and preprocessing procedures. To evaluate the performance, we leverage the widely used Area Under the ROC Curve (AUC) metric for offline evaluation.

3.1.2 Compared Baselines. To comprehensively verify the effectiveness of our model, we compare it with four classes of existing models, categorized by the degree of feature interactions: (1) First-Order (only uses individual features): Logistic Regression (LR); (2) Second-Order (modeling pair-wise feature interactions): Factorized Machine (FM) and AFM [28]; (3) Third-order (modeling triple-wise feature interactions): CrossNet (two-layer) [24], CrossNetV2 (two-layer) [25], and CIN (two-layer) [13]; (4) High-order (modeling interactions among more features): DCN, DCNV2 [25], DeepFM [10], AutoInt [22], xDeepFM [13], and SAM [5].

3.1.3 Implementation Details. We implement all the studied models based on FuxiCTR, an open-source CTR prediction library[2]. We follow the same experimental settings in [4] for fair comparisons. All baselines are trained with the Adam optimizer, where the learning rate is 0.001, the batch size is 4096, the embedding dimension is 10, and the numbers of MLP hidden units are [400, 400, 400]. We adopt two *FINAL* blocks with two factorized interaction layers ($K = 2, N = 2$). The source code of our model will be released publicly after acceptance.

3.2 Performance Comparison

Table 1 presents the evaluation results on four datasets, from which we have the following findings. First, LR performs the worst performance on all datasets, which shows the necessity of feature interaction modeling in CTR prediction. Second, methods that can model higher-order feature interactions tend to achieve better performance, which is intuitive because more complicated feature relatedness can be considered. Since *FINAL* is especially strong in modeling high-order feature interactions, it achieves the best performance on all datasets and its advantage is significant ($p < 0.05$ in t-test). This validates the effectiveness of *FINAL* in capturing sophisticated feature relations. Third, a dual-block *FINAL* model slightly outperforms a single-block one. This is probably because using multiple blocks helps learn diverse feature interaction information with different structures and initialized parameters. Fourth, self-knowledge distillation can further improve the performance of multi-block *FINAL* model. This further shows the complementarity of knowledge encoded in different blocks and using knowledge distillation to fuse them can better guide *FINAL* block learning.

[2]https://fuxictr.github.io

Table 1: Performance of different models.

Class	Model	Criteo	Avazu	MovieLens	Frappe
First-Order	LR	78.86	75.16	93.42	93.56
Second-Order	FM	80.22	76.13	94.34	96.71
	AFM	80.44	75.74	94.72	96.97
Third-Order	CrossNet	79.47	75.45	93.85	94.19
	CrossNetV2	81.10	76.05	95.83	97.16
	CIN	80.96	76.26	96.02	97.76
High-Order	DCN	81.39	76.47	96.87	98.39
	DCNV2	81.42	76.54	96.91	98.45
	DeepFM	81.38	76.48	96.85	98.42
	AutoInt	81.40	76.50	96.93	98.49
	xDeepFM	81.39	76.49	96.97	98.45
	SAM	81.31	76.32	96.31	98.01
Ours	$\text{FINAL}_{\text{single}}$	81.44	76.61	97.06	98.52
	$\text{FINAL}_{\text{dual}}$	81.45	76.64	97.11	98.83
	FINAL(2B)+KD	**81.56**	**76.67**	**97.20**	**98.95**

3.3 Compatibility Analysis

Our *FINAL* block is a plug-and-play module that can boost the performance of various deep CTR models. To demonstrate the compatibility of *FINAL* block, we introduce it as a drop-in replacement of the MLP block in four popular deep CTR models (i.e., MLP, DeepFM, xDeepFM, and DCN), and the results are shown in Table 2. We observe that *FINAL* block improves popular deep CTR models consistently. This verifies that *FINAL* indeed captures useful clues that are neglected by these models. Since *FINAL* is independent of backbone architecture, it is a flexible component used to empower various CTR prediction models in real-world systems.

Table 2: Compatibility study of *FINAL* block.

Model	Criteo	Avazu	MovieLens	Frappe
MLP	81.36	76.30	96.78	98.33
MLP+FINAL	**81.47**	**76.60**	**97.09**	**98.50**
DeepFM	81.38	76.48	96.85	98.42
DeepFM+FINAL	**81.43**	**76.57**	**97.12**	**98.53**
xDeepFM	81.39	76.49	96.97	98.45
xDeepFM+FINAL	**81.43**	**76.56**	**97.11**	**98.51**
DCN	81.39	76.47	96.87	98.39
DCN+FINAL	**81.48**	**76.58**	**96.94**	**98.48**

3.4 Online Evaluation

We have deployed *FINAL* in multiple commercial scenarios held by our enterprise, driven by its significant performance improvement and low latency overhead. In this section, we take two representative scenes to show its superiority.

3.4.1 News feed recommendation. we perform online evaluation in our commercial news recommendation scenario where millions of daily active users consume digital news articles. The online A/B test lasts for a month from September 25th to October 25th, 2022. For online serving, we split 5% of the whole traffic as the experimental group, which includes over 300k active users. We compare our method against a well-crafted baseline model. The online results in 30 consecutive days are summarized in Fig. 3. Our model shows consistent online CTR improvements during the evaluation period, where the average CTR improvement is 3.17%. The additional online inference latency is increased by 22.22%, which is acceptable in our system. The experimental results demonstrate the effectiveness of *FINAL* in feed recommendation.

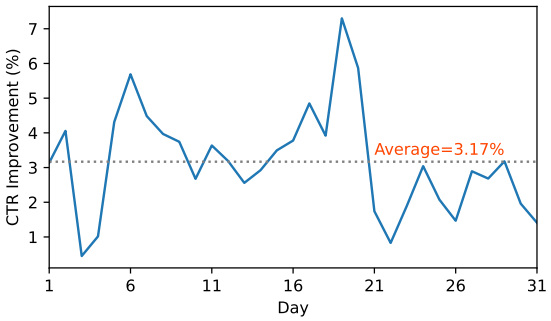

Figure 3: Improvements in one month online A/B testing.

3.4.2 Online advertisement display. Online advertising needs to predict CTR and Post-click conversion rate (CVR) simultaneously [18, 27]. In our ad display scenarios, the conversion corresponds to the events, such as installing apps, submitting registration information, user retention, etc. Multi-task learning (MTL) is a common solution for joint CTR and CVR estimation [16, 29]. In general, MTL adopts a model with shared-bottom structures, where the parameters of the bottom embedding layers are shared across tasks [17, 23, 26]. Then, an MLP module is applied to learn feature interactions from the shared bottom and make predictions for specific tasks. We use *FINAL* to replace MLP in this MTL framework for comparison. For online serving, we randomly select 5% of users as the experimental group, which receives ad recommendations by the *FINAL*-enhanced model. The control group with other 5% of users is served by the baseline MTL model. The online A/B test results of 7 consecutive days show a 5.52% overall CVR gain. The results validate the effectiveness of *FINAL* in online advertisement.

4 CONCLUSION

In this paper, we propose a universal architecture for high-order feature interaction modeling, named *FINAL*. It can achieve ultra-high feature interaction degrees at an exponential speed without conventional layer stacking by using a hierarchical structure to simulate the fast computation process of exponentiation. In addition, we propose to use multiple *FINAL* blocks to capture diverse feature interaction patterns, and we use self-distillation to exchange and fuse their inter-block knowledge to improve the overall prediction performance. Extensive experiments on four benchmark datasets and multiple online products show the effectiveness and generality of our *FINAL* method.

5 ACKNOWLEDGMENT

We gratefully acknowledge the support of MindSpore[3], which is a new deep learning computing framework.

[3]https://www.mindspore.cn

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 *WSDM*. ACM, 46–54.

[2] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. *TOIS* (2020).

[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 *DLRS@RecSys*. 7–10.

[4] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive factorization network: Learning adaptive-order feature interactions. In *AAAI*, Vol. 34. 3609–3616.

[5] Yuan Cheng and Yanbo Xue. 2021. Looking at CTR Prediction Again: Is Attention All You Need?. In *SIGIR*. ACM, 1279–1287.

[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *RecSys*. ACM, 191–198.

[7] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. 2021. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *ICML*. PMLR, 2793–2803.

[8] Ruili Feng, Kecheng Zheng, Yukun Huang, Deli Zhao, Michael Jordan, and Zheng-Jun Zha. 2022. Rank diminishing in deep neural networks. *arXiv preprint arXiv:2206.06072* (2022).

[9] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *IJCV* 129 (2021), 1789–1819.

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI*. 1725–1731.

[11] Boris Hanin. 2018. Which neural net architectures give rise to exploding and vanishing gradients? *NeurIPS* 31 (2018).

[12] Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. Fi-gnn: Modeling feature interactions via graph neural networks for ctr prediction. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 539–548.

[13] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In *KDD*. ACM, 1754–1763.

[14] Xiaoliang Ling, Weiwei Deng, Chen Gu, Hucheng Zhou, Cui Li, and Feng Sun. 2017. Model Ensemble for Click Prediction in Bing Search Ads. In *WWW*. ACM, 689–698.

[15] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction. In *KDD*. 2636–2645.

[16] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Why I like it: multi-task learning for recommendation and explanation. In *RecSys*. ACM, 4–12.

[17] 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 *KDD*. ACM, 1930–1939.

[18] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire Space Multi-Task Model: An Effective Approach for Estimating Post-Click Conversion Rate. In *SIGIR*. ACM, 1137–1140.

[19] Junwei Pan, Jian Xu, Alfonso Lobos Ruiz, Wenliang Zhao, Shengjun Pan, Yu Sun, and Quan Lu. 2018. Field-weighted factorization machines for click-through rate prediction in display advertising. In *WWW*. 1349–1357.

[20] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-Based Neural Networks for User Response Prediction. In *ICDM*. IEEE, 1149–1154.

[21] Steffen Rendle, Walid Krichene, Li Zhang, and John R. Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In *RecSys*. ACM, 240–248.

[22] 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 *CIKM*. ACM, 1161–1170.

[23] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In *RecSys*. ACM, 269–278.

[24] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *ADKDD*. ACM, 12:1–12:7.

[25] Ruoxi Wang, Rakesh Shivanna, Derek Z Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed H Chi. 2020. DCN-M: Improved deep & cross network for feature cross learning in web-scale learning to rank systems. *arXiv preprint arXiv:2008.13535* (2020).

[26] Penghui Wei, Weimin Zhang, Zixuan Xu, Shaoguo Liu, Kuang-chih Lee, and Bo Zheng. 2021. AutoHERI: Automated Hierarchical Representation Integration for Post-Click Conversion Rate Estimation. In *CIKM*. ACM, 3528–3532.

[27] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 2020. Entire Space Multi-Task Modeling via Post-Click Behavior Decomposition for Conversion Rate Prediction. In *SIGIR*. ACM, 2377–2386.

[28] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: learning the weight of feature interactions via attention networks. In *IJCAI*. 3119–3125.

[29] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-scale Causal Approaches to Debiasing Post-click Conversion Rate Estimation with Multi-task Learning. In *WWW*. ACM, 2775–2781.

[30] Chenxu Zhu, Bo Chen, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2021. AIM: Automatic Interaction Machine for Click-Through Rate Prediction. *TKDE* (2021).

[31] Jieming Zhu, Jinyang Liu, Weiqi Li, Jincai Lai, Xiuqiang He, Liang Chen, and Zibin Zheng. 2020. Ensembled CTR Prediction via Knowledge Distillation. In *CIKM*. ACM, 2941–2958.