Deep Position-wise Interaction Network for CTR Prediction

Jianqiang Huang, Ke Hu, Qingtao Tang, Mingjian Chen, Yi Qi, Jia Cheng, Jun Lei
Meituan
{huangjianqiang,huke05,tangqingtao,chenmingjian,qiyi02,jia.cheng.sh,leijun}@meituan.com

ABSTRACT

Click-through rate(CTR) prediction plays an important role in online advertising and recommender systems. In practice, the training of CTR models depends on click data which is intrinsically biased towards higher positions since higher position has higher CTR by nature. Existing methods such as actual position training with fixed position inference and inverse propensity weighted training with no position inference alleviate the bias problem to some extend. However, the different treatment of position information between training and inference will inevitably lead to inconsistency and sub-optimal online performance. Meanwhile, the basic assumption of these methods, i.e., the click probability is the product of examination probability and relevance probability, is oversimplified and insufficient to model the rich interaction between position and other information. In this paper, we propose a **D**eep **P**osition-wise **I**nteraction **N**etwork (DPIN) to efficiently combine all candidate items and positions for estimating CTR at each position, achieving consistency between offline and online as well as modeling the deep non-linear interaction among position, user, context and item under the limit of serving performance. Following our new treatment to the position bias in CTR prediction, we propose a new evaluation metrics named PAUC (position-wise AUC) that is suitable for measuring the ranking quality at a given position. Through extensive experiments on a real world dataset, we show empirically that our method is both effective and efficient in solving position bias problem. We have also deployed our method in production and observed statistically significant improvement over a highly optimized baseline in a rigorous A/B test.

CCS CONCEPTS

• **Information systems** → **Sponsored search advertising**;

KEYWORDS

CTR Prediction; Position-wise Interaction; Position Bias

ACM Reference Format:
Jianqiang Huang, Ke Hu, Qingtao Tang, Mingjian Chen, Yi Qi, Jia Cheng, Jun Lei. 2021. Deep Position-wise Interaction Network for CTR Prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463117

1 INTRODUCTION

In cost-per-click (CPC) advertising systems, advertisers are charged for every ad click, and advertisements are ranked by the eCPM (effective cost per mile), which is the product of click-through rate (CTR) and bid price. Hence,CTR prediction is a core task and has a direct impact in the final revenue and user experience.

In general, the implicit feedback collected from the abundant user clicks is used to train the CTR models. However, position bias exists in implicit feedback and hurt the model performance. The position bias happens as users tend to clicks on items in higher position regardless of the items’ actual relevance so that the CTR declines rapidly with the display position[4, 8].

Since position signal greatly impacts the CTR prediction, there has been a great deal of work on solving position bias problem. Modeling position as a feature in neural network[9, 13, 22] is widely adopted in industrial applications due to its simplicity and effectiveness, in which actual position feature is added in the wide part of neural network during offline training and a default position value will be used during online inference. To avoid using position feature online, Guo et al.[8] proposed a PAL framework to conduct online inference without position information. Additional, many works use inverse propensity weighting(IPW)[1, 2, 10, 12, 16, 19–21] to assign different weights to samples during model training. Other methods like knowledge distillation[14], adversarial neural networks[15], pairwise training[3, 11] have also been proposed.

Most of the above methods usually assume that the click Bernoulli variable C depends on two hidden Bernoulli variables E and R:

$$p(C = 1|u, c, i, k) = p(E = 1|k, [s])p(R = 1|u, c, i), \qquad (1)$$

where $p(C = 1|u, c, i, k)$ represents the probability that a user u click on the item i of k-th position in a search context c. We define the context as a collection of real-time request information including query, search time, location, etc. $p(E = 1|k, [s])$ represents the probability that position k is examined, and $[s]$ is the subset of context. Most methods commonly assume that the examination probability depends only on position k, i.e., $[s] = \emptyset$. $p(R = 1|u, c, i)$ represents the probability that the item is relevant to the user in the context. The assumption is used to eliminate position bias by estimating $p(E = 1|k, [s])$ implicitly or explicitly(modeling position as a feature in the wide part of neural network can be regarded as an implicit approach). And the relevance probability $p(R = 1|u, c, i)$ is obtained as the predicted CTR.

The different treatment of position information between training and inference will lead to inconsistency. Furthermore, different users usually have different browsing habits. The examination assumption is oversimplified and the position bias might be related to user, context and item.

In this paper, we propose a **D**eep **P**osition-wise **I**nteraction **N**etwork (DPIN) to model $CTR_k^j = p(C = 1|u, c, i, k)$ effectively and efficiently, where CTR_k^j is the predicted CTR of j-th candidate

SIGIR ’21, July 11–15, 2021, Virtual Event, Canada
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463117

item at k-th position for a specific request. The order of items is determined by maximizing $\sum CTR_k^j bid^j$, which is achieved by a greedy algorithm that selects a most valuable item from top position to bottom position. The contribution of this paper is summarized as follows:

- We propose to employ a shallow position-wise combination module with non-linear interaction in DPIN. Given a large number of combinations between candidate items and positions, this module is able to predict all the combinations' CTRs in parallel. This achieves consistency and greatly improves model performance.
- DPIN is the first method, as far as we know, to model user interests in candidate positions. We apply a deep position-wise interaction module to effectively represents deep non-linear interaction among position, user and context.
- A new evaluation metrics named PAUC (position-wise AUC) is proposed for measuring performance in solving position bias problem. We conduct extensive experiments on a real world dataset to show empirically that DPIN is both effective and efficient. Online A/B test is also deployed to demonstrate that DPIN has a significant improvement over a highly optimized baseline.

2 DEEP POSITION-WISE INTERACTION NETWORK

In this section, we introduce our DPIN. As shown in Figure 1, the DPIN is composed of three modules, which are base module for handling J items, deep position-wise interaction module for handling K positions and position-wise combination module for combining J items and K positions. By this network, CTRs for all candidates at each position can be predicted under the limit of serving performance. We will describe the three modules in detail.

2.1 Base Module

Similar to most deep CTR models[5–7, 18, 23, 24], we adopt the structure of embedding and MLP (Multiple Layer Perception) as our base module. For a request, the base module regards an user, a context and J candidate items as inputs, then the representation of j-th item with request information $\mathbf{r_j^{item}}$ is given via:

$$\mathbf{u} = Concat(E(u_1), ..., E(u_m)), \tag{2}$$

$$\mathbf{c} = Concat(E(c_1), ..., E(c_n)), \tag{3}$$

$$\mathbf{i_j} = Concat(E(i_1^j), ..., E(i_o^j)), \tag{4}$$

$$\mathbf{r_j^{item}} = MLP(Concat(\mathbf{u}, \mathbf{c}, \mathbf{i_j})), \tag{5}$$

where $\{u_1, ..., u_m\}$, $\{c_1, ..., c_n\}$,$\{i_1^j, ..., i_o^j\}$ are user features set, context features set and j-th item features set respectively. $E(\cdot) \in \mathbb{R}^d$ means the embedding which transforms sparse feature into learnable d-dimensional dense vector. Then, the concatenated vector is fed as input of MLP with $ReLU(x) = max(0, x)$ activation function.

2.2 Deep Position-wise Interaction Module

A brute force method in base module with $O(JKC)$ time complexity for CTR predictions of K positions is unacceptable due to the large time complexity $O(C)$, which is the inference latency of an item at a position. Therefore, we propose a deep position-wise interaction module paralleled with base module for learning deep non-linear interaction, in which item information is excluded.

In this module, the user behavior sequence at each position is retrieved independently for position-wise interest aggregation to eliminate position bias of lifelong sequence. Then, position-wise non-linear interaction among position, context and user are employed. Finally, transformer[17] is adopted for deep interaction among different positions.

Position-wise Interest Aggregation. We represent the user behavior sequence at k-th position as $B_k = \{b_1^k, b_2^k, ..., b_L^k\}$, in which $b_l^k = [v_l^k, c_l^k]$ stands for l-th behavior record. v_l^k is the l-th clicked item(item id, item category are included) and c_l^k is the context(query, location, hour of day, day of week are included) at which the interaction happened. Behavior embedding $\mathbf{b_l^k}$ can be obtained via:

$$\mathbf{b_l^k} = Concat(E(v_1^{k_l}), ..., E(v_o^{k_l}), E(c_1^{k_l}), ..., E(c_n^{k_l}), E(dif^{k_l})), \tag{6}$$

where $\{v_1^{k_l}, ..., v_o^{k_l}\}$, $\{c_1^{k_l}, ..., c_n^{k_l}\}$ are features set of v_l^k, c_l^k respectively and dif^{k_l} is the time difference between the behavior and current context.

The aggregation of behavior sequence at k-th position $\mathbf{b_k}$ is calculated by a context-aware attention for extracting user interests related to current context $\mathbf{c}$, which can be formulated as follows:

$$a_l^k = Concat(\mathbf{b_l^k}, \mathbf{c})\mathbf{W_a} + b_a, \tag{7}$$

$$\mathbf{b_k} = \sum_{l=1}^{L} \frac{\exp(a_l^k)}{\sum_{i=1}^{L} \exp(a_i^k)} \mathbf{b_l^k}, \tag{8}$$

where $\mathbf{W_a}$, b_a are learnable parameters.

Position-wise Non-linear Interaction. A fully connect with ReLU activation function is employed for non-linear interaction among position, context and user as follows:

$$\mathbf{v_k} = ReLU(Concat(E(k), \mathbf{c}, \mathbf{b_k})\mathbf{W_v} + \mathbf{b_v}), \tag{9}$$

where $\mathbf{W_v}, \mathbf{b_v}$ map concatenated vector to d_{model} dimension.

Transformer Block. If $\mathbf{v_k}$ is regarded as a non-linear interaction representation of k-th position, user interests at other positions will be lost. Hence the transformer is adopted for interaction among different positions. Same as [17], we donate the input of transformer as $\mathbf{Q} = \mathbf{K} = \mathbf{V} = Concat(\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_K}) \in \mathbb{R}^{K \times d_{model}}$. Multi-head self-attention can be formulated as follows:

$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(\mathbf{head_1}, ..., \mathbf{head_h})\mathbf{W^O}, \tag{10}$$

$$\mathbf{head_i} = Attention(\mathbf{QW_i^Q}, \mathbf{KW_i^K}, \mathbf{VW_i^V}), \tag{11}$$

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{QK^T}}{\sqrt{d_k}})\mathbf{V}, \tag{12}$$

where $\mathbf{W_i^Q}, \mathbf{W_i^K}, \mathbf{W_i^V} \in \mathbb{R}^{d_{model} \times d_k}, \mathbf{W^O} \in \mathbb{R}^{d_{model} \times d_{model}}$ are the parameter matrices. And $d_k = d_{model}/h$ is the dimension of each head. The output of multi-head self-attention is fed as the input of position-wise feed-forward network for non-linear transformation. Residual connections and layer normalization are adopted successively. And a stack of N transformer blocks is employed for deep interaction.

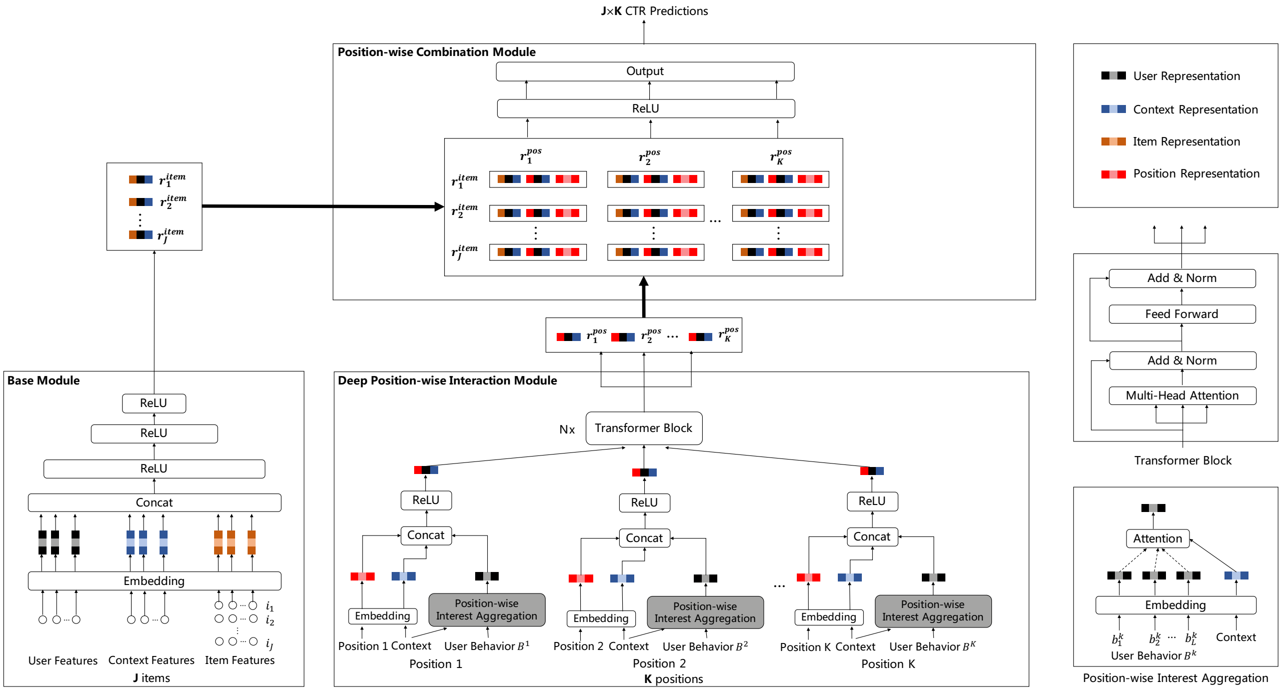

Figure 1: The structure of Deep Position-wise Interaction Network.

Finally, the representation of k-th position with request information $\mathbf{r}_\mathbf{k}^\mathbf{pos}$ is given by the proposed deep position-wise interaction module.

2.3 Position-wise Combination Module

The purpose of the position-wise combination module is to predict the CTR of each item at each position by combining J items and K positions. A non-linear interaction among $\mathbf{r}_\mathbf{j}^\mathbf{item}$, $\mathbf{r}_\mathbf{k}^\mathbf{pos}$ and $E(k)$ is used to learn non-linear relationship among user, context, item and position. The CTR of j-th item at k-th position CTR_k^j can be calculated as follows:

$$CTR_k^j = \sigma(ReLU(Concat(\mathbf{r}_\mathbf{j}^\mathbf{item}, \mathbf{r}_\mathbf{k}^\mathbf{pos}, E(k))\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + b_2), \quad (13)$$

where $\mathbf{W}_1, \mathbf{b}_1$ are the parameters of non-linear interaction layer. $\mathbf{W}_2, b_2$ are the parameters of output layer. $\sigma(\cdot)$ is sigmoid function.

Finally, the model can be trained using stochastic gradient descent algorithm with actual position feature, and cross-entropy is used as the loss function.

3 EXPERIMENT

In this section, we evaluate the model performance and serving performance of the proposed DPIN. We describe the experimental settings and experimental results in detail.

3.1 Experimental Settings

Datasets. A four-week ad impression log collected from a sponsored search advertising system in a shopping application is used to train the CTR models. We evaluate our methods on two test sets, which are collected from regular traffic and top-k randomized traffic the next day. The top-k randomized traffic is suitable for position bias evaluation since it excludes the impact of relevance recommendations. The number of impressions is about 10 million a day, and 5% of traffic is the randomized traffic.

Metrics. We use AUC (Area Under ROC) as one of our evaluation metrics. To evaluate the model performance of our methods for position bias, we propose PAUC (Position-wise AUC) as another evaluation metrics, which is calculated as follows:

$$PAUC = \frac{\sum_{k=1}^{K} \#impression_k PAUC@k}{\sum_{k=1}^{K} \#impression_k}, \quad (14)$$

where $\#impression_k$ is the number of impressions at k-th position, and $PAUC@k$ is the AUC of impressions at k-th position. PAUC measures the relevance ranking quality at every position, ignoring the impact of position bias.

Compared Methods. Our baseline is a highly optimized DIN[24] model, in which a large number of attributes and hand-crafted features are added. The number of features is up to 241 and embedding dimension k is 8 for each feature. The hidden sizes of MLP are 1024, 512, 128. And the hidden size of the non-linear interaction layer in position-wise combination module is 128. The length of users' behaviors is truncated to 300. And the number of positions K is truncated to 25, which accounts for most of the online traffic. In order to compare different methods fairly, we ensure that all input information and parameters setting of common module are consistent. We conducted experiments with the following methods:

DIN. Position is not used in this model.

DIN+PosInWide. The method models position feature in the wide part of neural network and first position is used for evaluation.

DIN+PAL. PAL method is adopted based on DIN.

DIN+ActualPosInWide. The method use actual position for evaluation based on DIN+PosInWide.

DIN+Combination. Position-wise combination module is employed based on DIN. Actual position is used for evaluation.

DPIN-Transformer. Transformer is not used in DPIN.

DPIN. There is the proposed DPIN model, in which a tow-layer transformer is adopted and $d_{model} = 64, h = 2$ for self-attention.

DPIN+ItemAction. We add deep position-wise interaction module before the MLP layer of the base module, and candidate item information is introduced in position-wise interest aggregation and position-wise non-linear interaction. The experiment is the upper bound of model performance in our methods, but the serving performance is unacceptable.

3.2 Offline Evaluations

Table 1: Offline experimental results of compared methods on the regular and randomized test sets.

	Regular		*Randomized*	
Model	AUC	PAUC	AUC	PAUC
DIN	0.7818	0.7090	0.7836	0.7223
DIN+PosInWide	0.7696	0.7109	0.7725	0.7239
DIN+PAL	0.7735	0.7128	0.7763	0.7254
DIN+AcutalPosInWide	0.7928	0.7109	0.7938	0.7239
DIN+Combination	0.7970	0.7172	0.7985	0.7294
DPIN-Transformer	0.7961	0.7148	0.7984	0.7283
DPIN	**0.7994**	**0.7216**	**0.8015**	**0.7350**
DPIN+ItemAction	0.7999	0.7223	0.8019	0.7356

Table 1 illustrates experimental results of the compared methods on both regular and randomized test sets. We first analyze the differences between different methods on the regular traffic. Compared with DIN, the DIN+PosInWide and DIN+PAL have a performance degradation on AUC but an improvement on PAUC, which shows that both methods effectively alleviate position bias but lead to inconsistency between offline and online. The DIN+AcutalPosInWide solves the inconsistency by introducing actual position during evaluation, which is achievable by position-wise combination module. But modeling position in the wide part leads to the position feature is only a bias, which can not improve PAUC. By employing our proposed position-wise combination module, the DIN+Combination has 2.74% gain on AUC and 0.63% gain on PAUC compared with DIN+PosInWide, achieving consistency and alleviate position bias further, which shows the position bias is not independent. Furthermore, DPIN models deep non-linear interaction among position, context and user, and eliminates position bias existed in the user sequence by position-wise method, which has 0.24% gain on AUC and 0.44% gain on PAUC compared with DIN+Combination. The effect of DPIN-Transformer explains that it is necessary to adopt transformer for interaction among different positions. And the comparison between DPIN and DPIN+ItemAction shows that DPIN is close to the brute force method on both AUC and PAUC. As can be seen finally, the DPIN has 2.98% gain on AUC and 1.07% gain on PAUC relative to the DIN+PosInWide, which is a baseline in the advertising system online.

In order to ensure that our method can learn the position bias instead of overfitting the selection bias of the system, we further evaluate our methods on randomized traffic. The results show that the differences between the different methods on both regular and randomized traffic is consistent.

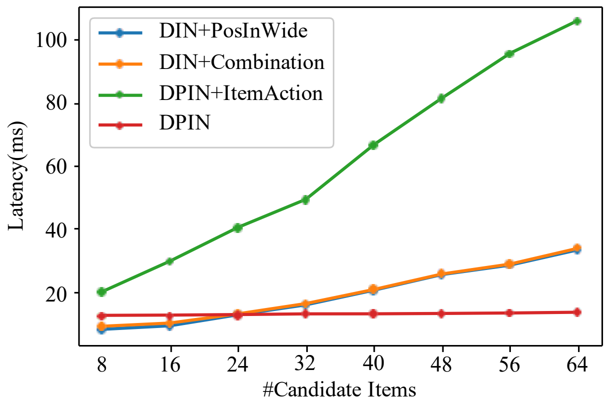

Figure 2: Serving latency changes with the number of candidate items in different methods.

3.3 Serving Performance

We retrieve some requests with different candidate item numbers from the dataset to measure serving performance. As shown in Figure 2, the serving latency of position-wise combination module is negligible compared to the DIN model since user sequence operation has a large proportion of latency. The serving latency of DPIN increases slowly as the number of items increases since the deep position-wise interaction module has nothing to do with items. Compared with DIPIN+ItemAction, the DPIN has a great improvement in serving performance with little damage to model performance, which shows that our proposed method is both effective and efficient.

3.4 Online Evaluations

Online A/B test was conducted in the sponsored search advertising system from 2021-01-08 to 2021-01-22. For the control group, 10% of users are randomly selected and presented with recommendation generated by DIN+PosInWide. For the experimental group, 10% of users are presented with recommendation generated by DPIN. The A/B test shows that the proposed DPIN has improved CTR by 2.25% and RPM (Revenue Per Mille) by 2.15% compared with baseline. For now, DPIN has been deployed online and serves the main traffic, which contributes a significant business revenue growth.

4 CONCLUSIONS

In this paper, a novel model Deep Position-wise Interaction Network (DPIN) is proposed to mitigate position bias, which efficiently combine all candidate items and positions for estimating CTR at each position, achieving consistency between offline and online. At the same time, the deep non-linear interaction among position, context and user is available by the model. In order to evaluate our method in position bias problem, we propose a new evaluation metrics PAUC and the offline experiments show that the proposed DPIN outperforms the compared methods efficacy and efficiency. For now, the DPIN is deployed in a sponsored search advertising system and serving the main traffic.

REFERENCES

[1] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating position bias without intrusive interventions. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 474–482.

[2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 385–394.

[3] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2212–2220.

[4] 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. *arXiv preprint arXiv:2010.03240* (2020).

[5] 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.

[6] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. *arXiv preprint arXiv:1905.06482* (2019).

[7] 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).

[8] Huifeng Guo, Jinkai Yu, Qing Liu, Ruiming Tang, and Yuzhou Zhang. 2019. PAL: a position-bias aware learning framework for CTR prediction in live recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 452–456.

[9] Malay Haldar, Prashant Ramanathan, Tyler Sax, Mustafa Abdool, Lanbo Zhang, Aamir Mansawala, Shulin Yang, Bradley Turnbull, and Junshuo Liao. 2020. Improving Deep Learning For Airbnb Search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2822–2830.

[10] Ziniu Hu, Yang Wang, Qu Peng, and Hang Li. 2019. Unbiased lambdamart: an unbiased pairwise learning-to-rank algorithm. In *The World Wide Web Conference*. 2830–2836.

[11] Jiarui Jin, Yuchen Fang, Weinan Zhang, Kan Ren, Guorui Zhou, Jian Xu, Yong Yu, Jun Wang, Xiaoqiang Zhu, and Kun Gai. 2020. A deep recurrent survival model for unbiased ranking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 29–38.

[12] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 781–789.

[13] Xiaoliang Ling, Weiwei Deng, Chen Gu, Hucheng Zhou, Cui Li, and Feng Sun. 2017. Model ensemble for click prediction in bing search ads. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 689–698.

[14] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 831–840.

[15] John Moore, Joel Pfeiffer, Kai Wei, Rishabh Iyer, Denis Charles, Ran Gilad-Bachrach, Levi Boyles, and Eren Manavoglu. 2018. Modeling and Simultaneously Removing Bias via Adversarial Neural Networks. *arXiv preprint arXiv:1804.06909* (2018).

[16] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for selection bias in learning-to-rank systems. In *Proceedings of The Web Conference 2020*. 1863–1873.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[18] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.

[19] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 115–124.

[20] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 610–618.

[21] Bowen Yuan, Yaxu Liu, Jui-Yang Hsia, Zhenhua Dong, and Chih-Jen Lin. 2020. Unbiased Ad click prediction for position-aware advertising systems. In *Fourteenth ACM Conference on Recommender Systems*. 368–377.

[22] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 43–51.

[23] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2018. Deep Interest Evolution Network for Click-Through Rate Prediction. *arXiv preprint arXiv:1809.03672* (2018).

[24] 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.