Deep Page-Level Interest Network in Reinforcement Learning for Ads Allocation

Guogang Liao*
Meituan
Beijing, China
liaoguogang@meituan.com

Xiaowen Shi*
Meituan
Beijing, China
shixiaowen03@meituan.com

Ze Wang†
Meituan
Beijing, China
wangze18@meituan.com

Xiaoxu Wu
Meituan
Beijing, China
wuxiaoxu04@meituan.com

Chuheng Zhang‡
IIIS, Tsinghua University
Beijing, China
zhangchuheng123@live.com

Yongkang Wang
Meituan
Beijing, China
wangyongkang03@meituan.com

Xingxing Wang
Meituan
Beijing, China
wangxingxing04@meituan.com

Dong Wang
Meituan
Beijing, China
wangdong07@meituan.com

ABSTRACT

A mixed list of ads and organic items is usually displayed in feed and how to allocate the limited slots to maximize the overall revenue is a key problem. Meanwhile, user behavior modeling is essential in recommendation and advertising (e.g., CTR prediction and ads allocation). Most previous works only model point-level positive feedback (i.e., click), which neglect the page-level information of feedback and other types of feedback. To this end, we propose Deep Page-level Interest Network (DPIN) to model the page-level user preference and exploit multiple types of feedback. Specifically, we introduce four different types of page-level feedback, and capture user preference for item arrangement under different receptive fields through the multi-channel interaction module. Through extensive offline and online experiments on Meituan food delivery platform, we demonstrate that DPIN can effectively model the page-level user preference and increase the revenue.

CCS CONCEPTS

• **Information systems** → **Computational advertising**; **Online advertising**; **Electronic commerce**.

KEYWORDS

Ads Allocation, Reinforcement Learning, User Behavior Modeling

ACM Reference Format:
Guogang Liao, Xiaowen Shi, Ze Wang, Xiaoxu Wu, Chuheng Zhang, Yongkang Wang, Xingxing Wang, and Dong Wang. 2022. Deep Page-Level Interest Network in Reinforcement Learning for Ads Allocation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3477495.3531847

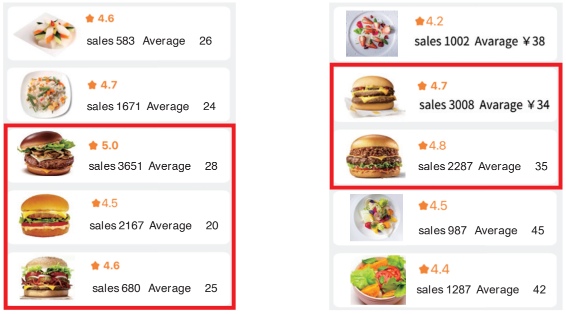
1 INTRODUCTION

Ads and organic items are mixed together and displayed to users in e-commerce feed nowadays [3, 5, 15] and how to allocate the limited slots to maximize the overall revenue has become a key problem [7, 11, 16]. Since the feed is presented to the user in a sequence, recent ads allocation strategies model the problem as Markov Decision Process (MDP) [10] and solve it using reinforcement learning (RL) [2, 6, 16, 18, 19]. For instance, Xie et al. [14] propose a hierarchical RL-based framework to first decide the type of the item to present and then determine the specific item for each slot. Liao et al. [6] proposes CrossDQN which takes the crossed state-action pairs as input and allocates the slots in one page at a time.

User behavior modeling, which focuses on learning the intent representation of user interest, is widely introduced in recommendation and advertising scenarios (e.g., CTR prediction and ads allocation) [9, 12, 20]. Most previous works on user behavior modeling [12, 17, 20] only model user interest using positive feedback (e.g., click) while neglect other types of feedback, which may result in an inaccurate approximation of user interest. Xie et al. [13] model both positive and negative feedback and achieve better performance. However, they only model point-level feedback, which ignore the page-level information of the feedback (e.g., mutual influence among items in one page). Fan et al. [1] introduce page-wise feedback sequence but still face three major limitations. Firstly, it would be better to match historical page-level feedback with the target page rather than the target item. Secondly, as shown in Figure 1, different users may have different preferences on receptive field when browsing, which means users may pay attention to the

[*] Equal contribution. Listing order is random.
[†] Corresponding author.
[‡] This work was done when Chuheng Zhang was an intern in Meituan.

SIGIR ’22, July 11–15, 2022, Madrid, Spain.
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531847



(a) User A focuses on three items in a page (b) User B focuses on two items in a page

Figure 1: Different users may have different preferences on receptive field when browsing.

mutual influence among items within different ranges. Thirdly, they ignore other types of page-level feedback (e.g., pull-down, leave).

To address these limitations, we present a method named Deep Page-level Interest Network (DPIN)[1] to model the page-level user preference for ads allocation and exploit multiple types of feedback. Specifically, we construct four page-level behavior sequence (i.e., page-level order, click, pull-down and leave) and use the Multi-Channel Interaction Module (MCIM) to model page-level user preference. In MCIM, we first use multiple convolution kernels with different sizes to extract the information of different receptive fields on the page. Nextly, we conduct Intra-Page Attention Unit (IPAU) to capture the mutual influence among items within different ranges. Subsequently, we design the Inter-Page Interaction Unit (IPIU) to calculate the correlation between target page and page-level sequences and denoise implicit feedback (i.e., unclick and pull down, hereinafter referred to as pull-down) by sequence interaction.

We have conducted several offline experiments and evaluated our approach on real-world food delivery platform. The experimental results show that the introduction of page-level historical behavior and the modeling of page-level user preference can significantly improve the platform revenue. This is a meaningful attempt in modeling page-level user preference on ads allocation.

2 PROBLEM FORMULATION

Items are displayed to the user page by page in our scenario and each page consists of K slots. We handle the ads allocation problem for each page sequentially, which can be formulated as a MDP ($\mathcal{S}$, $\mathcal{A}$, r, P, γ). The elements are defined as follows:

- **State space** $\mathcal{S}$. A state $s \in \mathcal{S}$ consists of the candidate items (i.e., the ads list and the organic items list which are available on current step t), the user’s profile features (e.g., age, gender), the context features (e.g., order time, order location) and four types of user’s page-level historical behavior sequences (i.e, page-level order, click, pull-down and leave).
- **Action space** $\mathcal{A}$. An action $a \in \mathcal{A}$ determines whether to display an ad on each slot of the page, which can be formulated as follows:

$$a = (x_1, x_2, \ldots, x_K),$$
$$\text{where } \ x_k = \begin{cases} 1 & \text{display an ad in the } k\text{-th slot} \\ 0 & \text{otherwise} \end{cases}, \ \forall k \in [K]. \tag{1}$$

  In our scenario, the order of the items within ads sequence and organic items sequence remain unchanged during the allocation.
- **Reward** r. After the system takes an action, a user browses the page and gives feedback. The reward is calculated based on the feedback which consists of ads revenue r^{ad} and service fees r^{fee}:

$$r(s, a) = r^{\text{ad}} + r^{\text{fee}}. \tag{2}$$

- **Transition probability** P. $P(s_{t+1}|s_t, a_t)$ is defined as the state transition probability from s_t to s_{t+1} after taking the action a_t in the t-th page. When the user pulls down, the state s_t transits to the state of next page s_{t+1}. The items selected by a_t will be removed from the state on the next step s_{t+1} since seeing the same items in different pages may affect user experience. If the user no longer pulls down, the transition terminates.
- **Discount factor** γ. The discount factor $\gamma \in [0, 1]$ balances the short-term and long-term rewards.

Given the MDP formulated as above, the objective is to find an ads allocation policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ to maximize the total reward.

3 METHODOLOGY

As shown in Figure 2, DPIN mainly consists of three main components, i.e. Input & Embedding Layer, Multi-Channel Interaction Module (MCIM) and Multi-Layer Perceptron (MLP). Next, we will detail each part.

3.1 Input & Embedding Layer

The input information consists of five parts: context features, user profile features, four page-level historical behavior sequences, candidate ads and organic items sequences, and candidate actions. Similar to Liao et al. [6], we cross the candidate ads and organic items sequences according to the action to form the target page arrangement $t(s, a)$. The page-level historical behavior sequences for the user include four types: page-level sequence of order, click, pull-down, and leave. We use "page-level" to refer to the present items on current page when the user’s behavior occurred. We concatenate the position feature and feedback category feature for each item to improve the sequence representation.

We use embedding layers to extract the embeddings from raw inputs. The embedding matrix for page-level information is denoted as $\mathbf{E} \in \mathbb{R}^{K \times d}$, where K is the number of presented items on a page and d is the dimension of embedding. We denote the embeddings for page-level order sequence, click sequence, pull-down sequence, leave sequence, target page, the user profile, the context as $\{\mathbf{E}_i^{\text{o}}\}_{i=1}^{N_{\text{o}}}$, $\{\mathbf{E}_i^{\text{c}}\}_{i=1}^{N_{\text{c}}}$, $\{\mathbf{E}_i^{\text{p}}\}_{i=1}^{N_{\text{p}}}$, $\{\mathbf{E}_i^{\text{l}}\}_{i=1}^{N_{\text{l}}}$, $\mathbf{E}^t$, $\mathbf{e}^{\text{u}}$, and $\mathbf{e}^{\text{c}}$ respectively, where the subscript i denotes the index within the sequence and N_{o}, N_{c}, N_{p} and N_{l} are the length of corresponding behavior sequences.

[1] The code and data example are publicly accessible at https://github.com/princewen/DPIN

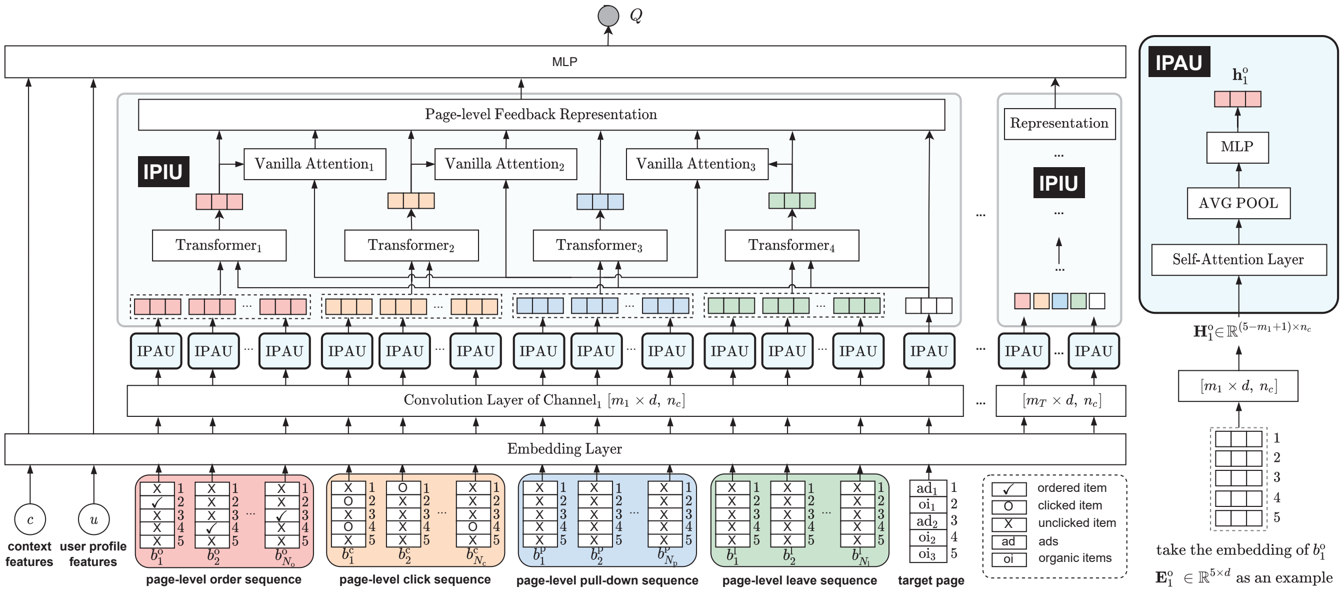


Figure 2: The structure of Deep Page-level Interest Network. The features are first input into the Embedding Layer. Then the embeddings of four page-level sequences and target page are input into the multi-channel interaction module to generate representations. The output multiple page-level feedback representations are concatenated with the embeddings of context and user profile to predict the value Q.

3.2 Multi-Channel Interaction Module (MCIM)

Different arrangements of displayed items on a page make different influence on user behaviors. Accordingly, we propose MCIM to model page-level user preferences. Different channels can capture user preference for item arrangement under different receptive fields through three parts: Convolution Layer (CL), Intra-Page Attention Unit (IPAU) and Inter-Page Interaction Unit (IPIU). Next, we will use the structure of a channel to introduce each part.

3.2.1 *Convolution Layer.* Each page-level embedding matrix $\mathbf{E}$ is first input into the convolution layer to extract the local field information of the page:

$$\mathbf{H}_1 = f_{n_c}^{m\times d}(\mathbf{E}), \tag{3}$$

where n_c is the number of convolution kernels, m is the size of receptive fields, and $\mathbf{H}_1 \in \mathbb{R}^{(K-m+1)\times n_c}$ is the output.

3.2.2 *Intra-Page Attention Unit.* Then the matrix are input into self-attention layer which uses the scaled dot-product attention:

$$\mathbf{H}_2 = \mathrm{SDPA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{soft\,max}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{n_c}})\mathbf{V}, \tag{4}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ represent query, key, and value, respectively. d denotes feature dimension of each feature. Here, query, key and value are transformed linearly from $\mathbf{H}_1$, as follows:

$$\mathbf{Q} = \mathbf{H}_1\mathbf{W}^Q, \mathbf{K} = \mathbf{H}_1\mathbf{W}^K, \mathbf{V} = \mathbf{H}_1\mathbf{W}^V, \tag{5}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{n_c\times n_c}$. Then $\mathbf{H}_2$ are input into a MLP to generate the page-level representation:

$$\mathbf{h} = \mathrm{MLP}_1\Big(\text{avg pool}\big(\mathbf{H}_2\big)\Big). \tag{6}$$

3.2.3 *Inter-Page Interaction Unit.* Xie et al. [13] have proved that historical behaviors which are more relevant to the target item can provide more information for the model's predict. Therefore, we use the multi-head self-attention to calculate the interactions between target page and page-level sequences. Take the order sequence as example. We combine the representation of target page with the page-level representations of order sequence to form the input matrix $\mathbf{B}_\mathrm{o} = \{\mathbf{h}_t, \mathbf{h}_1^\mathrm{o}, \cdots, \mathbf{h}_{N_\mathrm{o}}^\mathrm{o}\}$. Similar to Xie et al. [13], we then use multi-head self-attention to generate the interacted page-level order sequence representation and formulate the result as $\mathbf{z}^\mathrm{o}$.

Notice that, the length of page-level implicit feedback (i.e., pull-down) sequence is obviously longer than the other three, which can be noisy to some extent [13], since the items exposed are carefully selected by ranking strategies or user may scroll too fast to notice them. Accordingly, we use the other three sequences to discern the page-level arrangement that user may or may not prefer in the pull-down feedback sequence. Take $\mathbf{z}^\mathrm{o}$ as example, the denoised representation of pull-down feedback $\mathbf{z}^{\mathrm{p(o)}}$ is calculated as:

$$\begin{aligned} w_{p_i}^\mathrm{o} &= \mathrm{MLP}_2\Big(\mathbf{h}_i^\mathrm{p}||\mathbf{z}^\mathrm{o}||(\mathbf{h}_i^\mathrm{p}\odot\mathbf{z}^\mathrm{o})||(\mathbf{h}_i^\mathrm{p}-\mathbf{z}^\mathrm{o})\Big), \\ w_{p_i}^\mathrm{o} &= \frac{\exp(w_{p_i}^\mathrm{o})}{\sum_{j=1}^{N^\mathrm{p}}\exp(w_{p_j}^\mathrm{o})}, \\ \mathbf{z}^{\mathrm{p(o)}} &= \sum_{i=1}^{N_\mathrm{p}} w_{p_i}^\mathrm{o}\mathbf{h}_i^\mathrm{p}. \end{aligned} \tag{7}$$

We concatenate the extracted representations as the page-level feedback representation in current channel. The outputs of different channels will be concatenated together as follows:

$$\mathbf{c}_i = \mathbf{z}^{\mathrm{o}}||\mathbf{z}^{\mathrm{c}}||\mathbf{z}^{\mathrm{p}}||\mathbf{z}^{\mathrm{l}}||\mathbf{z}^{\mathrm{p(o)}}||\mathbf{z}^{\mathrm{p(c)}}||\mathbf{z}^{\mathrm{p(l)}}||\mathbf{h}^{\mathrm{t}}, \\ \mathbf{e}_{\mathrm{MCIM}} = \mathbf{c}_1||\mathbf{c}_2||...||\mathbf{c}_T. \tag{8}$$

3.3 Optimization Objective

We concatenate the output of MCIM with the embeddings of context and user profile to predict the value Q through an MLP:

$$Q(s,a) = \mathrm{MLP}_3\Big(\mathbf{e}_{\mathrm{MCIM}}||\mathbf{e}_c||\mathbf{e}_u\Big). \tag{9}$$

For each iteration, we sample a batch of transitions B from the offline dataset and update the agent using gradient back-propagation w.r.t. the loss [8]:

$$L(B) = \frac{1}{|B|}\sum_{(s,a,r,s')\in B}\Big(r + \gamma \max_{a'\in\mathcal{A}} Q(s',a') - Q(s,a)\Big)^2. \tag{10}$$

4 EXPERIMENTS

We will evaluate our DPIN through offline and online experiments in this section. In offline experiments, we compare our method with existing state-of-the-art baselines and analyze the role of different units and different page-level behavior sequences. In online part, we compare our method with the previous strategy deployed on Meituan food delivery platform using an online A/B test.

4.1 Experimental Settings

4.1.1 Dataset. Since there are no public datasets for ads allocation problem, we collect the dataset by running an exploratory policy on Meituan food delivery platform during January 2022. The dataset contains 12,411,532 requests, 1,732,492 users, 358,394 ads and 710,937 organic items. We use the user's request within 30 days to obtain the user's four types of page-level sequences (page-level order, click, pull-down and leave). The average length of the four sequences is 4.63, 10.10, 39.4 and 12.68, respectively.

4.1.2 Evaluation Metrics. We evaluate with the ads revenue $R^{\mathrm{ad}} = \sum r^{\mathrm{ad}}$, the service fee $R^{\mathrm{fee}} = \sum r^{\mathrm{fee}}$. See the definition in Section 2.

4.1.3 Hyperparameters. We apply a gird search for the hyperparameters. The length of each sequence is truncated (or padded) to 10, the number of channel is 5, the hidden layer sizes of all MLPs are $(128, 64, 32)$, the τ is 0.9, the learning rate is 10^{-3}, the optimizer is Adam [4] and the batch size is 8,192.

4.2 Offline Experiment

In this section, we train our method with offline data and evaluate the performance using an offline estimator. Through extended engineering, the offline estimator models the user preference and aligns well with the online service.

4.2.1 Baselines. We compare our method with the following representative RL-based dynamic ads slots methods:

- **HRL-Rec** divides the integrated recommendation into two levels of tasks and solves using hierarchical reinforcement learning.
- **DEAR** designs a deep Q-network architecture to determine three related tasks jointly, i.e., i) whether to insert an ad, and if yes, ii) the optimal ad and iii) the optimal location to insert.
- **CrossDQN** takes the crossed state-action pair as input and allocates slots in one page at a time. It designs some units (e.g., MCAU) to optimize the mutual impact of the items.
- **CrossDQN & DIN** introduces point-level order sequence into CrossDQN . The sequence is modeled with DIN [21].
- **CrossDQN & DFN** introduces point-level order, click, pull-down, leave sequences into CrossDQN and sequences are modeled with DFN [13].
- **CrossDQN & RACP** introduces page-level order, click, pull-down, leave sequences into CrossDQN and sequence is modeled with RACP [13].

Table 1: The experimental results. Each experiment is presented in the form of mean ± standard deviation. The improvement means the improvements of our method across the best baselines.

model	R^{ad}	R^{fee}
HRL-Rec	0.1114 (±0.0002)	0.9485 (±0.0255)
DEAR	0.1119 (±0.0003)	0.9545 (±0.0198)
CrossDQN	0.1149 (±0.0005)	0.9761 (±0.0063)
CrossDQN&DIN	0.1150 (±0.0006)	0.9789 (±0.0082)
CrossDQN&DFN	0.1153 (±0.0003)	0.9824 (±0.0050)
CrossDQN&RACP	0.1157 (±0.0003)	0.9836 (±0.0100)
Our method	**0.1181 (±0.0003)**	**1.0105 (±0.0102)**
- w/o CL	0.1161 (±0.0007)	0.9883 (±0.0033)
- w/o IPAU	0.1167 (±0.0005)	0.9872 (±0.0098)
- w/o IPIU	0.1160 (±0.0002)	0.9843 (±0.0067)
- w/o MCIM	0.1151 (±0.0006)	0.9781 (±0.0059)
- w/o $\{\mathbf{E}^{\mathrm{p}}\}$&$\{\mathbf{E}^{\mathrm{l}}\}$	0.1163 (±0.0005)	0.9999 (±0.0096)
- w/o $\{\mathbf{E}^{\mathrm{p}}\}$&$\{\mathbf{E}^{\mathrm{l}}\}$&$\{\mathbf{E}^{\mathrm{c}}\}$	0.1158 (±0.0003)	0.9957 (±0.0189)
Improvement	1.7%	2.2%

4.2.2 Performance Comparison. We present the experimental results in Table 1. Compared with all these baselines, our method achieves strongly competitive performance on both the ads revenue and the service fee. Specifically, our method improves over the best baseline w.r.t. R^{ad} and R^{fee} by 1.7% and 2.2% separately. The superior performance of our method justifies that effectiveness of modeling page-level user preference through multiple types of page-level behavior sequences.

4.2.3 Ablation Study. To verify the impact of our designs, we study six ablated variants of our method and have the following findings: i) The performance gap between w/ and w/o CL verifies the effectiveness of modeling user preference for item arrangement under different receptive fields. ii) The performance gap between w/ and w/o IPIU verifies the effectiveness of calculating the correlation between target page and page-level sequences. iii) The performance gap between w/ and w/o $\{\mathbf{E}^{\mathrm{p}}\}$&$\{\mathbf{E}^{\mathrm{l}}\}$ and the performance gap w/

and w/o $\{\mathbf{E}^{\mathrm{p}}\}\&\{\mathbf{E}^{\mathrm{l}}\}\&\{\mathbf{E}^{\mathrm{c}}\}$ verify the effectiveness of utilizing multiple types of page-level feedback.

4.2.4 Hyperparameter Analysis. We analyze the sensitivity of the number of channels in our method. The experimental results show that model achieve better performance as the number of channels increases, especially the number of channels grows from 1 to 3.

4.3 Online Results

We compare DPIN with CrossDQN and both strategies are deployed on Meituan food delivery platform through online A/B test for 14-days. Each group contains millions of users to validate the results at a statistically significant level. As a result, we find that R^{ad} and R^{fee} increase by 1.5% and 1.7%, with the average increased cost of 3.8 milliseconds for online inference, which demonstrates that DPIN can greatly increase the platform revenue.

5 CONCLUSIONS

In this paper, we propose a method for page-level historical behavior sequence modeling on ads allocation problem. Specifically, we introduce four different types of page-level feedback (i.e., page-level order, click, pull-down, leave) as input, and capture user preference for item arrangement under different receptive fields through the multi-channel interaction module. Practically, both offline experiments and online A/B test have demonstrated the superior performance and efficiency of our method.

REFERENCES

[1] Zhifang Fan, Dan Ou, Yulong Gu, Bairan Fu, Xiang Li, Wentian Bao, Xin-Yu Dai, Xiaoyi Zeng, Tao Zhuang, and Qingwen Liu. 2022. Modeling Users' Contextualized Page-wise Feedback for Click-Through Rate Prediction in E-commerce Search. (2022).

[2] Jun Feng, H. Li, Minlie Huang, Shichen Liu, Wenwu Ou, Zhirong Wang, and Xiaoyan Zhu. 2018. Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning. *Proceedings of the 2018 World Wide Web Conference* (2018).

[3] A. Ghose and Sha Yang. 2009. An Empirical Analysis of Search Engine Advertising: Sponsored Search in Electronic Markets. *Manag. Sci.* 55 (2009), 1605–1622.

[4] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[5] Xiang Li, Chao Wang, Bin Tong, Jiwei Tan, Xiaoyi Zeng, and Tao Zhuang. 2020. Deep Time-Aware Item Evolution Network for Click-Through Rate Prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 785–794.

[6] Guogang Liao, Ze Wang, Xiaoxu Wu, Xiaowen Shi, Chuheng Zhang, Yongkang Wang, Xingxing Wang, and Dong Wang. 2021. Cross DQN: Cross Deep Q Network for Ads Allocation in Feed. *arXiv preprint arXiv:2109.04353* (2021).

[7] Aranyak Mehta. 2013. Online Matching and Ad Allocation. *Found. Trends Theor. Comput. Sci.* 8 (2013), 265–368.

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[9] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2671–2679.

[10] Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning.* Vol. 135. MIT press Cambridge.

[11] B. Wang, Zhaonan Li, Jie Tang, Kuo Zhang, Songcan Chen, and Liyun Ru. 2011. Learning to Advertise: How Many Ads Are Enough?. In *PAKDD.*

[12] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep multi-interest network for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 2265–2268.

[13] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2021. Deep feedback network for recommendation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence.* 2519–2525.

[14] Ruobing Xie, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. 2021. Hierarchical Reinforcement Learning for Integrated Recommendation. In *Proceedings of AAAI.*

[15] Jinyun Yan, Zhiyuan Xu, Birjodh Tiwana, and Shaunak Chatterjee. 2020. Ads Allocation in Feed via Constrained Optimization. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 3386–3394.

[16] Weiru Zhang, Chao Wei, Xiaonan Meng, Yi Hu, and Hao Wang. 2018. The whole-page optimization via dynamic ad allocation. In *Companion Proceedings of the The Web Conference.* 1407–1411.

[17] Keke Zhao, Xing Zhao, Qi Cao, and Linjian Mo. 2021. A Non-sequential Approach to Deep User Interest Model for CTR Prediction. *arXiv preprint arXiv:2104.06312* (2021).

[18] Xiangyu Zhao, Changsheng Gu, Haoshenglun Zhang, Xiwang Yang, Xiaobing Liu, Hui Liu, and Jiliang Tang. 2021. DEAR: Deep Reinforcement Learning for Online Advertising Impression in Recommender Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 750–758.

[19] Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. 2020. Jointly learning to recommend and advertise. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 3319–3327.

[20] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[21] 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.