Task Adaptive Multi-learner Network for Joint CTR and CVR Estimation

Xiaofan Liu[2*], Qinglin Jia[1*], Chuhan Wu[1], Jingjie Li[1], Quanyu Dai[1],
Lin Bo[3], Rui Zhang[4†], Ruiming Tang[1†]
[1]Noah's Ark Lab, Huawei [2]Beijing University of Posts and Telecommunications
[3]Renmin University of China [4]ruizhang.info
{jiaqinglin2,wuchuhan,lijingjie1,daiquanyu,tangruiming}@huawei.com,liuxiaofan@bupt.edu.cn
bolin20@ruc.edu.cn,rayteam@yeah.net

ABSTRACT

CTR and CVR are critical factors in personalized applications, and many methods jointly estimate them via multi-task learning to alleviate the ultra-sparsity of conversion behaviors. However, it is still difficult to predict CVR accurately and robustly due to the limited and even biased knowledge extracted by the single model tower optimized on insufficient conversion samples. In this paper, we propose a task adaptive multi-learner (TAML) framework for joint CTR and CVR prediction. We design a hierarchical task adaptive knowledge representation module with different experts to capture knowledge in different granularities, which can effectively exploit the commonalities between CTR and CVR estimation tasks meanwhile keeping their unique characteristics. We apply multiple learners to extract data knowledge from various views and fuse their predictions to obtain accurate and robust scores. To facilitate knowledge sharing across learners, we further perform self-distillation that uses the fused scores to teach different learners. Thorough offline and online experiments show the superiority of TAML in different Ad ranking tasks, and we have deployed it in Huawei's online advertising platform to serve the main traffic.

CCS CONCEPTS

• **Information systems → Computational advertising**; • **Computing methodologies → Multi-task learning**.

KEYWORDS

Multi-task Learning, Recommender System, CVR Estimation

ACM Reference Format:
Xiaofan Liu, Qinglin Jia, Chuhan Wu, Jingjie Li, Quanyu Dai, Lin Bo, Rui Zhang, Ruiming Tang. 2023. Task Adaptive Multi-learner Network for Joint CTR and CVR Estimation. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 4, 2023, Austin, Texas, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3543873.3584653

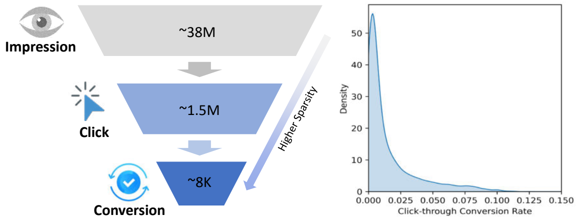

Figure 1: The progressively increased sparsity of events as well as the CTCVR distribution in Ali-CCP.

1 INTRODUCTION

Online advertising is essential to customer acquisition, which typically follows a sequential event pattern of "impression → click → conversion" [22]. For example, in the app promotion scenario, users may first click an Ad if they are attracted by its content, and then decide whether to trigger a conversion by installing the corresponding App or submitting registration forms[1]. To target users' interest and increase advertisers' return on investment (ROI), click-through rate (CTR) prediction and post-click conversion rate (CVR) prediction are essential tasks for personalized Ad ranking systems [4, 10, 17, 19], both of which have decisive impacts on advertisers' bid under various mainstream pricing rules like optimized cost-per-click (OCPC) [12].

Due to the cascaded process of customer acquisition, the amount of positive feedback is progressively reduced (left Fig. 1), which leads to a severe sparsity problem of conversion events and thereby differentiates the characteristics of CTR and CVR model learning[10, 14]. However, conventional methods usually separately estimate CTR and CVR with similar techniques (e.g., factorization machines [5, 11] and deep neural networks (DNN) [3, 7]), which may suffer from the sparsity of conversion samples [10, 15, 19]. Thus, there has been an increasing trend in industrial applications to jointly estimate CTR and CVR via multi-task learning (MTL) [8–10, 20, 21]. For example, Ma et al. [10] propose ESMM to optimize both CTR and CTCVR (click-through conversion rate) tasks on post-view samples, where the CTCVR score is the multiplication of predicted CTR and CVR scores. Ma et al. [9] propose to use shared experts controlled by gating mechanisms below task-specific networks, which aims to model the commonalities among tasks. However, due to the ultra-sparsity of conversion events, the knowledge learned by these methods for CVR estimation may still be rather limited and even biased [2, 13]. Concerning the diverse CTCVRs of different items

[*]Equal contribution.
[†]The corresponding author.

WWW '23, April 30–May 4, 2023, Austin, Texas, USA

ACM ISBN 978-1-4503-9419-2/23/04…$15.00
https://doi.org/10.1145/3543873.3584653

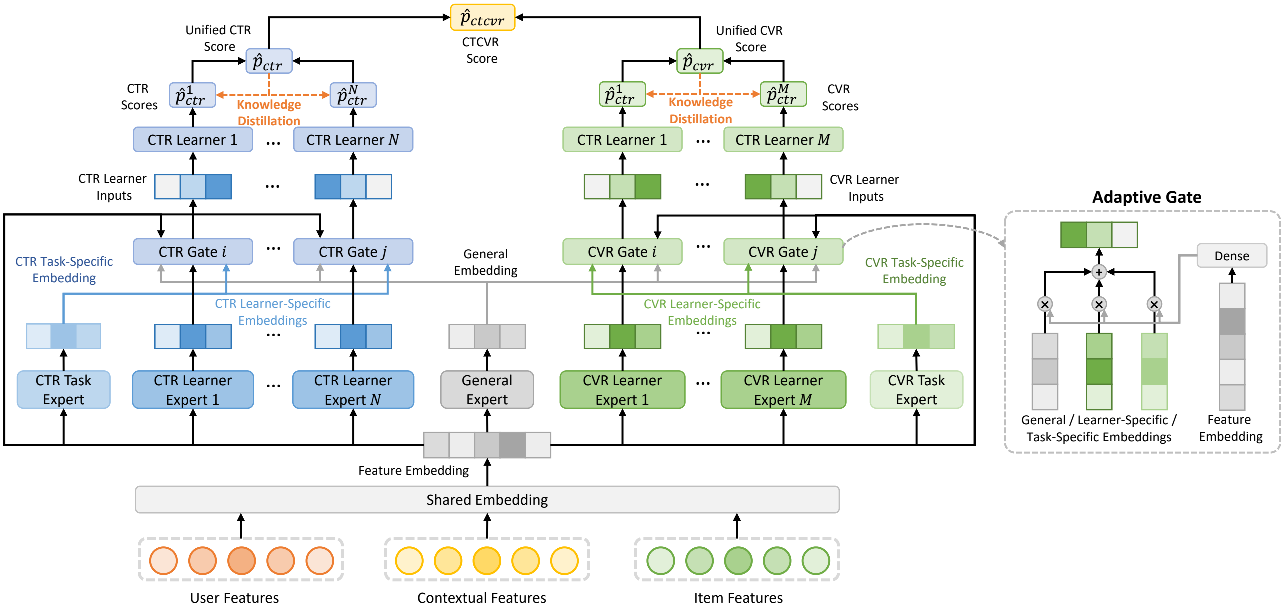

Figure 2: Overview of TAML. The whole framework is trained on the post-view space.

(right Fig. 1), it may also be difficult for an insufficiently-optimized single model tower to generate accurate and robust predictions.

In this paper, we propose a **T**ask **A**daptive **M**ulti-**l**earner (TAML)[1] framework for joint CTR and CVR estimation, which employs a hierarchical task adaptive knowledge representation module with different experts to capture multi-grained knowledge, and uses multiple learners to exploit and exchange knowledge from multiple views to facilitate accurate and robust prediction. Specifically, the hierarchical task adaptive knowledge representation module uses three groups of experts to respectively capture task-agnostic, task-specific, and even finer-grained learner-specific knowledge. On the top of these experts, each task has multiple learners to adapt to its characteristics and make predictions, which are further synthesized into the final output. To facilitate knowledge exchange among learners, we apply self-distillation[18, 23] by using the fused output as a virtual teacher to teach them to make accurate and robust predictions. Both online and offline experiments show the advantage of TAML over baselines, and it has been serving the main traffic of Huawei's online advertising system for mobile app promotion.

2 METHODOLOGY

2.1 Overall Framework

Suppose the input features of the model are X , which are looked up by a shared embedding layer to get the feature embedding vector $\mathbf{x}$. Then it is input to a hierarchical task adaptive knowledge representation (HTAKR) module, which extracts knowledge from the feature embedding in three granularities, including a general level, a task-specific level, and a finer-grained learner-specific level. The outputs from this module will be aggregated by a set of adaptive gates, which adaptively control the importance of the three types of knowledge according to the feature embedding. Each gate corresponds to a task-specific learner, which further encodes knowledge from different views and makes predictions for its task. The prediction scores from multiple learners are aggregated into a unified score as the final output of its task, which also serves as the virtual teacher to teach learners of this task via self-distillation to exchange cross-learner knowledge. We finally multiply the probability results of CTR and CVR tasks to get the CTCVR score $\hat{p}_{ctcvr}$.

2.2 HTAKR Module

To fully model task-agnostic and task-specific knowledge, we design a hierarchical task-adaptive knowledge representation module on the top of the embedding layer. It includes general experts to learn universal knowledge across tasks, task-specific experts to learn task-sensitive knowledge, and learner-specific experts to further cover task-aware knowledge in different aspects. To adaptively fuse the knowledge extracted by the three groups of experts, an adaptive gate is used to generate the input embedding for each learner from the outputs of general experts and experts of the corresponding task and learner, as shown in right Fig. 2. Let l represent the l-th learner and t represent the task $t \in \{ctr, cvr\}$, the input of each learner is denoted as $f^l_t(\mathbf{x})$. The adaptive gate uses a single-layer feed-forward network with Softmax function to assign weights to each expert and aggregate the inputs as follows:

$$g^l(\mathbf{x}) = \text{Softmax}\left(\mathbf{W}^l\mathbf{x}\right), \quad f^l(\mathbf{x}) = \sum_{i=1}^{k} g^l(\mathbf{x})_i e^l(\mathbf{x})_i, \qquad (1)$$

where $\mathbf{W}^l$ are parameters of the l-th learner, k is the total number of experts, $e^l(\mathbf{x})$ is the concatenation of the embeddings generated by the general experts, task t's experts, and l-th learner's expert.

[1] The MindSpore implementation is available at https://gitee.com/mindspore/models/tree/master/research/recommend/TAML.

2.3 Multi-Learner Network with Self-Distillation

The input of each learner $f^l(\mathbf{x})$ first passes through a multi-layer perceptron and gets the output logits, i.e., $logit_t^l = MLP(f_t^l(x))$. Then the output logits of each learner are aggregated by average as $logit_t = \sum_{l=1}^{n} \frac{1}{n} logit_t^l$, where $logit_t$ represents the ensemble results in the task t and n is the total number of learners in this task. The ensemble results are normalized by the *sigmoid* function, which is denoted as $\hat{p}_t$. To encourage knowledge sharing among learners of each task, we apply self-distillation to the muti-learner network. We set the fused results as the teacher and each learner as the student. Denote the normalized output of each learner as $\hat{p}_t^l$, the self-distillation loss functions in different tasks are formulated as follows:

$$\mathcal{L}_{self-kd_{ctr}} = -\sum_{l=1}^{n}[\hat{p}_{ctr}\log(\hat{p}_{ctr}^l) + (1-\hat{p}_{ctr})(1-\log(\hat{p}_{ctr}^l))], \quad (2)$$

$$\mathcal{L}_{self-kd_{cvr}} = -\sum_{l=1}^{m}[\hat{p}_{cvr}\log(\hat{p}_{cvr}^l) + (1-\hat{p}_{cvr})(1-\log(\hat{p}_{cvr}^l))]. \quad (3)$$

2.4 Joint Loss Optimization for TAML

Following [10], we use the cross-entropy loss function to supervise the CTR and CTCVR tasks as follows:

$$\mathcal{L}_{ctr} = -\frac{1}{N}\sum_{i=1}^{n}\left[y_{ctr}^i \log\left(\hat{p}_{ctr}^i\right) + \left(1 - y_{ctr}^i\right)\log\left(1 - \hat{p}_{ctr}^i\right)\right], \quad (4)$$

$$\mathcal{L}_{ctcvr} = -\frac{1}{N}\sum_{i=1}^{n}\left[y_{ctcvr}^i \log\left(\hat{p}_{ctcvr}^i\right) + \left(1 - y_{ctcvr}^i\right)\log\left(1 - \hat{p}_{ctcvr}^i\right)\right]. \quad (5)$$

where $\mathcal{L}_{ctr}$ and $\mathcal{L}_{ctcvr}$ are the loss functions of the CTR and CTCVR tasks, respectively. The final loss function $\mathcal{L}$ of the TAML framework consists of four parts, including the loss of two supervised tasks and the loss of two self-distillation auxiliary tasks:

$$\mathcal{L} = \mathcal{L}_{ctr} + \mathcal{L}_{ctcvr} + \alpha\mathcal{L}_{self-kd_{ctr}} + \beta\mathcal{L}_{self-kd_{cvr}} \quad (6)$$

where α and β are hyper-parameters, which are the weights of the self-distillation loss in CTR and CVR tasks.

3 EXPERIMENTS

3.1 Datasets

We use two datasets in the experiments. The first one is a public dataset named Alibaba Click and Conversion Prediction (Ali-CCP)[2]. It is collected from real-world traffic logs of the recommender systems in the Taobao platform. Following [16], we filter the features whose frequencies are less than 10 and divide 10% of the samples in the original training set for validation. The second one is a proprietary dataset (named Industrial) collected from Huawei's advertising platform. It contains 8 consecutive days of logged data from a browser search application. The feature set of this dataset is comprised of 12 fields, including user attributes, item information, and corresponding context. We use samples on days 1-7 for training and day 8 for test. The details of both datasets are listed in Table 1.

Table 1: Statistics of the two datasets.

Dataset	#User	#Item	#Impression	#Click	#Conversion
Ali-CCP	0.4M	4.3M	84M	3.4M	18k
Industrial	0.5M	67.6k	736M	3.7M	1.8M

3.2 Experimental Settings

In our experiments, we optimize the hyperparameters of baselines and our methods by grid search. We set up a four-layer deep neural network for all MLP-based methods, and the numbers of hidden units in each layer are [192, 64, 32, 1]. We use three experts in expert-based models, each of which is a one-layer perceptron with 64 hidden units. In our method, we use two learners for each task[3], and each learner is a three-layer perception with 64, 32, and 1 hidden units. The weights of self-distillation loss α and β are set to 0.1. For model training, Adam [6] is the optimizer with a learning rate of 1e-3, and the batch size is 2000. We use L2 regularization with the strength of 1e-6. ReLU is the activation function. In all offline experiments, we use ROC AUC of CTR, CVR, CTCVR tasks as the evaluation metric. We run five rounds of experiments for each method and record the average results.

Table 2: Model AUC on the Ali-CCP and Industrial datasets.

Model	Ali-CCP Dataset			Industrial Dataset		
	CTR	CVR	CTCVR	CTR	CVR	CTCVR
MLP	0.6034	0.6519	0.6207	0.8892	0.7569	0.9525
MMoE	0.6026	0.6703	0.6400	0.9051	0.8082	0.9533
PLE	0.6037	0.6712	0.6477	0.9037	0.8140	0.9539
ESMM	0.6030	0.6706	0.6485	0.9043	0.8162	0.9530
AITM	0.6035	0.6693	0.6518	0.9044	0.8112	0.9541
TAML	**0.6049**	**0.6813**	**0.6544**	**0.9072**	**0.8177**	**0.9557**

3.3 Offline Evaluation

We conduct extensive experiments on the Ali-CCP and industrial datasets by comparing TAML with the following baselines: (1) **MLP**, using separate multi-layer perceptrons to predict the target for each task; (2) **MMoE**: using multiple shared experts to extract common knowledge; (3) **PLE**: using both task-specific experts and shared experts to achieve progressive knowledge extraction; (4) **ESSM**: joint estimation of different tasks via probability multiplication; (5) **AITM**: transferring information from previous tasks to backward tasks and using a calibrator to constrain the prediction. The evaluation results are shown in Table 2, from which we have the following observations. In the two tasks of CVR and CTCVR, MLP shows low performance, which is due to the sparsity of conversion samples. Besides, MLP does not use the multi-task learning method to learn the CTR task, which leads to insufficient learning in the discrimination of impression samples in the testing data. PLE is better than MMoE in most tasks since the task-specific experts in PLE can independently assist in each task learning. PLE can learn the commonalities and differences between CTR and CVR

[2] https://tianchi.aliyun.com/datalab/dataSet.html?dataId=408

[3] Although increasing the number of learners can further improve performance, the computational complexity is also larger. Thus, we use two learners that can achieve the maximum relative improvement.

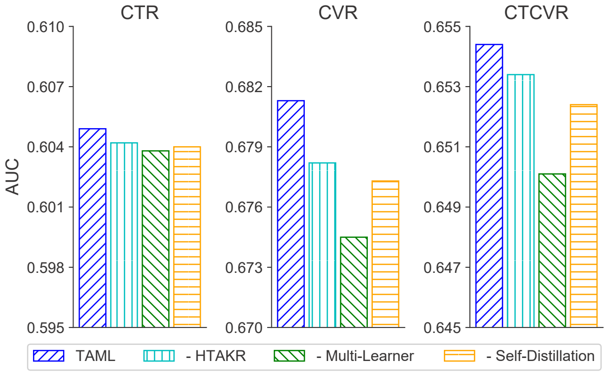

Figure 3: Ablation studies on core modules in TAML.

prediction and improves overall performance. These two methods achieve improvement to some degree, but they have not taken advantage of the sequential relationship between CTR and CTCVR tasks. ESMM implicitly models p_{cvr} at the top of the neural network by means of probabilistic information transfer. This entire space multi-task modeling method makes it perform better than MLP in both CVR and CTCVR tasks. However, ESMM has not considered the commonalities and differences between CTR and CVR prediction tasks at the bottom of the MLP. AITM uses the AIT and calibrator modules to improve CTCVR task with CTR task, but it cannot completely avoid the phenomenon of $p_{ctcvr} > p_{ctr}$ because the calibrator can only act as a regularizer rather than restricting it from the probability formula. In addition, this method does not explicitly or implicitly model the CVR task, which makes its low performance in the CVR task. TAML achieves state-of-the-art in all tasks, which is not only due to the improvement achieved by the Multi-Learner module learning knowledge from different views but also because of the self-distillation loss that encourages the knowledge exchange among learners. And the HTAKR module has better performance in extracting knowledge representation information for each task.

3.4 Online Evaluation

The proposed method and baseline MTL method were trained offline and regularly update from April 9 to April 21. Each pre-trained model is deployed in a single cluster to real-timely show advertising for mobile app promotion. For online serving, we randomly select 2% of the users as the experimental group that is recommended ads by TAML and another 2% of the users as the control group that is recommended ads by baseline MTL method. We use CVR and Effective Cost per Mille (eCPM) to evaluate the performance of the above deployed models.

The online A/B test results of consecutive 15 days show significant improvement of TAML over the baseline MTL model. Compared with the baseline MTL model, the conversion rate increases by 2.49%, and eCPM increases by 2.41%. It brings significant business revenue improvement with slight latency overload. Now, TAML has provided real-time prediction for the major traffic in our system.

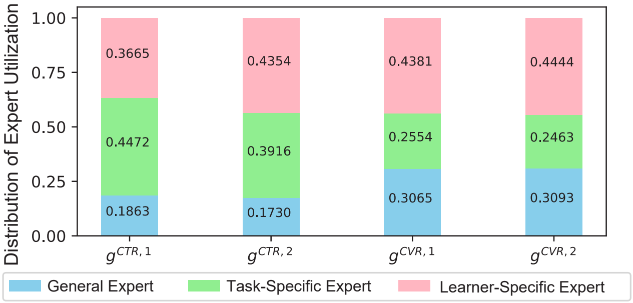

Figure 4: Expert weights in adaptive gates.

3.5 Ablation Study

To validate the effectiveness of each module in TAML, we conduct ablation studies on the Ali-CCP dataset. The experimental results are shown in Fig. 3. From the results, we find each module plays an important role in our method, and removing each of them will lead to performance degradation. Among them, the multi-learner mechanism has the most salient contribution. This is because using multiple learners can help encode more comprehensive and less biased knowledge. The self-distillation mechanism also has a notable contribution. It may be because using knowledge distillation to fuse the knowledge extracted from multiple learners is better than simply increasing model capacity. The HTAKR module also has some contributions. This shows that disentangling different types of knowledge is more suitable for multi-task learning.

3.6 Gating Weight Analysis

Here, we investigate the gating weights of the three types of experts on Ali-CCP, as shown in Fig. 4. We find different tasks have diverse gating weights. In the CTR task, general experts are less important than task-specific experts, while on the contrary in the CVR task. This is because the CTR task can provide rich supervision signals to assist CVR task in distinguishing a large number of negative impression samples so that the CTR task is relatively more helpful to the CVR task. The learner-specific experts play important roles in both CTR and CVR tasks. It shows the importance of fine-grained learner-specific knowledge in CTR and CVR prediction.

4 CONCLUSION

In this paper, we propose a task adaptive multi-learner (TAML) framework for joint CTR and CVR estimation. TAML uses a hierarchical task adaptive knowledge representation module to disentangle knowledge in different granularities, and compose them adaptively as the input of multiple learners. The learners of each task generate final predictions collaboratively, and they exchange their hidden knowledge via self-distillation. Experiments on both public and proprietary datasets show that TAML outperforms the previous state-of-the-art models on three important tasks CTR, CVR, and CTCVR. Meanwhile, online experiments on Huawei's advertising platform show that our model achieves notable online improvements, and it is deployed to serve the personalized Ad recommendation service for App promotion.

REFERENCES

[1] Narayan Bhamidipati, Ravi Kant, and Shaunak Mishra. 2017. A Large Scale Prediction Engine for App Install Clicks and Conversions. In *CIKM*. ACM, 167–175.

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

[4] Quanyu Dai, Haoxuan Li, Peng Wu, Zhenhua Dong, Xiao-Hua Zhou, Rui Zhang, Rui Zhang, and Jie Sun. 2022. A Generalized Doubly Robust Learning Framework for Debiasing Post-Click Conversion Rate Prediction. In *KDD*. ACM, 252–262.

[5] 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.

[6] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[7] 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.

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

[9] 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.

[10] 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.

[11] Yang Sun, Junwei Pan, Alex Zhang, and Aaron Flores. 2021. FM2: Field-matrixed Factorization Machines for Recommender Systems. In *WWW*. ACM, 2828–2837.

[12] Shulong Tan, Meifang Li, Weijie Zhao, Yandan Zheng, Xin Pei, and Ping Li. 2021. Multi-Task and Multi-Scene Unified Ranking Model for Online Advertising. In *IEEE Big Data*. 2046–2051.

[13] Yu Wang, Xin Xin, Zaiqiao Meng, Joemon M Jose, Fuli Feng, and Xiangnan He. 2022. Learning Robust Recommenders through Cross-Model Agreement. In *WWW*. ACM.

[14] 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.

[15] 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.

[16] Dongbo Xi, Zhen Chen, Peng Yan, Yinger Zhang, Yongchun Zhu, Fuzhen Zhuang, and Yu Chen. 2021. Modeling the Sequential Dependence among Audience Multi-step Conversions with Multi-task Learning in Targeted Display Advertising. In *KDD*. ACM, 3745–3755.

[17] Zixuan Xu, Penghui Wei, Weimin Zhang, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. UKD: Debiasing Conversion Rate Estimation via Uncertainty-Regularized Knowledge Distillation. In *WWW*. 2078–2087.

[18] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In *ICCV*.

[19] 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.

[20] Jiejie Zhao, Bowen Du, Leilei Sun, Fuzhen Zhuang, Weifeng Lv, and Hui Xiong. 2019. Multiple Relational Attention Network for Multi-task Learning. In *KDD*. ACM, 1123–1131.

[21] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed H. Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *RecSys*. ACM, 43–51.

[22] Han Zhu, Junqi Jin, Chang Tan, Fei Pan, Yifan Zeng, Han Li, and Kun Gai. 2017. Optimized Cost per Click in Taobao Display Advertising. In *KDD*. ACM, 2191–2200.

[23] 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.