Making the Full Model Adaptive: Multi-Level Domain Adaptation for Multi-Domain CTR Prediction

Qi Zhang*, Chuhan Wu†, Jieming Zhu†, Jingjie Li†, Qinglin Jia†, Ruiming Tang†, Rui Zhang‡, Liangbi Li*

*VMALL, Huawei Technologies Co. Ltd. †Huawei Noah's Ark Lab ‡ ruizhang.info

zhangqi193@huawei.com, wuchuhan@huawei.com, jiemingzhu@ieee.org, lijingjie1@huawei.com

ABSTRACT

Multi-domain CTR prediction helps a single recommender model serve multiple domains with awareness of their relatedness, and existing methods usually add domain-specific layers on a shared model to consider domain characteristics. However, different domains may have distinct feature spaces and importance, and the shared model cannot effectively unify them and may neglect useful domain relations. In this paper, we propose a multi-level domain adaptation method for multi-domain CTR prediction. It introduces domain awareness to many critical steps in CTR prediction, including feature embedding, feature selection, and feature representation, to better bridge and fuse multi-domain signals. Concretely, we maintain a set of meta-embeddings for each feature field and compose them into domain-aware feature embeddings. We then select them in a domain-aware way to promote informative features for different domains. Finally, we use a domain-adaptive router to select proper submodels from multiple candidates to learn domain-specific representations. Extensive experiments on both public and proprietary datasets validate the effectiveness of our method. Its online deployment also achieves notable improvements over well-crafted predecessors.

CCS CONCEPTS

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

KEYWORDS

Recommender system, CTR prediction, multi-domain recommendation, domain adaptation

ACM Reference Format:

Qi Zhang, Chuhan Wu, Jieming Zhu, Jingjie Li, Qinglin Jia, Ruiming Tang, Rui Zhang, Liangbi Li. 2024. Making the Full Model Adaptive: Multi-Level Domain Adaptation for Multi-Domain CTR Prediction. In *DLP '23: International Workshop on Deep Learning Practice for High-Dimensional Sparse Data, September 18–22, 2023, Singapore.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Click-through rate (CTR) prediction is critical for many personalized services, such as e-commerce recommendation and targeted advertising [23, 24, 27]. These services usually require to handle items across different domains to satisfy various user needs [5, 25].

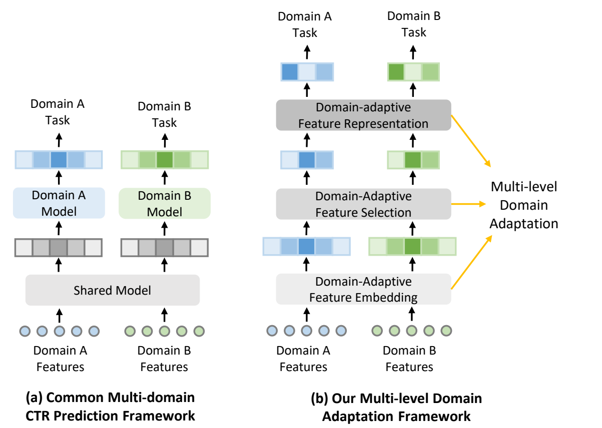

Figure 1: A comparison of two multi-domain CTR prediction frameworks.

Since domains can be divergent in their characteristics, it may be suboptimal to simply mix multi-domain data for model training [19]. Many methods train and maintain separate models for each domain [15], which is quite cumbersome and expensive in real-world scenarios with various domains [8, 20, 29].

To reduce the training and maintenance overhead of online recommender systems, multi-domain CTR prediction aims to handle multiple domains with a single model and exploit the inherent relatedness among domains [5, 17, 19, 21, 26, 30]. A common paradigm for multi-domain CTR prediction is first using a base model shared among different domains for feature representation and then building domain-specific models on it to adapt to domain characteristics [6], as shown in Fig. 1(a). For example, He et al. [6] proposed to use a shared bottom network to generate common feature representations for different domains and add domain-specific layers to consider domain characteristics. Sheng et al. [19] proposed to combine a shared model and domain-specific models via element-wise product to generate domain-aware feature representations. However, different domains usually have huge barriers in terms of their feature spaces and feature importance [1, 2, 9, 11, 14, 22], which may hinder the shared model from unifying multi-domain information and capturing domain commonalities.

In this paper, we propose a multi-level domain adaptation framework for multi-domain CTR prediction, named *AdaptiveCTR*. Instead of using a partially shared model across domains, our method is fully domain-adaptive in different critical steps of CTR prediction, including feature embedding, feature selection, and feature representation (Fig. 1(b)), which can better consider diverse domain characteristics and synthesize multi-domain information. Specifically, we first convert each raw feature into its meta-embeddings

DLP '23, September 18–22, 2023, Singapore

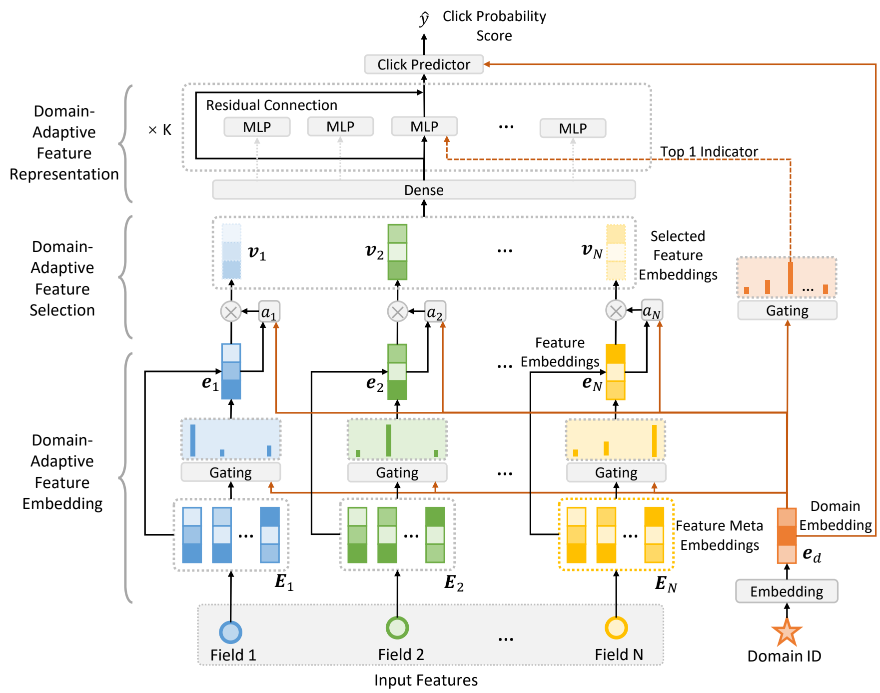

Figure 2: The architecture of *AdaptiveCTR*.

and compose them into domain-aware feature embeddings according to domain characteristics, where the same feature is represented differently in different domains. Next, we use domain information to guide feature selection to recognize salient features for each domain. Finally, we use multiple submodels to learn feature representations and employ a domain-adaptive router to select the results to get informative and domain-aware hidden representations for CTR prediction. We conduct experiments on a public dataset and a proprietary dataset collected from a commercial advertising platform, and the results validate the effectiveness of our method. We have deployed *AdaptiveCTR* in an online Ad system and achieved notable improvements over the previous base models.

2 OUR APPROACH

Here we introduce our *AdaptiveCTR* method in detail. Its overall framework is shown in Fig. 2. It receives a set of user and item features as the input, and outputs a click probability score for personalized recommendation. Our method mainly contains three levels of feature processing, including feature embedding, feature selection, and feature representation, which gradually converts raw features into informative hidden representations for CTR prediction. These steps are all domain-adaptive due to the guidance of domain information, thereby the model can fully consider domain characteristics to better union heterogeneous but relevant signals in different domains. The details of each step are introduced below.

2.1 Domain-adaptive Feature Embedding

Maintaining one embedding for each feature is a common practice in existing methods [3]. However, the same feature may evoke different implications and semantics in different domains, which is difficult to be condensed by a single embedding. Thus, we propose to maintain a set of meta-embeddings for each feature and compose them differently according to domain characteristics to generate domain-specific feature embeddings. Specifically, we denote the meta-embeddings of the i-th feature field as $\mathbf{E}_i \in \mathbb{R}^{d\times h}$, where h is the number of meta-embeddings and d is the embedding dimension. We use a domain-specific gating network to select a proper subset of meta-embeddings and further compose them into a unified feature embedding. Denote the embedding of domain ID as $\mathbf{e}_d$. It helps compute a gating score vector $\mathbf{g}_i$ for the meta-embeddings of the i-th field as follows:

$$\mathbf{g}_i = \mathrm{softmax}(\mathbf{U}_i\mathbf{z}_i + \alpha\mathbf{z}_i),\ \mathbf{z}_i = \mathrm{ReLU}(\mathbf{W}_i\mathbf{e}_d), \tag{1}$$

where $\mathbf{W}_i \in \mathbb{R}^{h\times d}$ and $\mathbf{U}_i \in \mathbb{R}^{h\times h}$ are parameters, α is a learnable factor that controls skip-connections. The output embedding $\mathbf{e}_i$ of i-th field is computed by $\mathbf{e}_i = \mathbf{E}_i\mathbf{g}_i$. In this way, feature embeddings are adaptively generated based on domain characteristics.

2.2 Domain-adaptive Feature Selection

Selecting salient features is essential for CTR prediction [12, 13]. Different domains may be divergent in their feature spaces, and even the same feature may yield disparate importance in different domains. Thus, we propose a domain-adaptive feature selection method to assign features different importance weights to highlight domain-sensitive features. Denote all feature embeddings as $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N]$. We compute a feature weighting vector $\mathbf{a}$ as:

$$\mathbf{a} = \mathrm{softmax}\left(\frac{(\mathbf{W}_K\mathbf{E})(\mathbf{W}_Q\mathbf{e}_d)}{\sqrt{d}}\right) \tag{2}$$

where $\mathbf{W}_Q$ and $\mathbf{W}_K$ are parameters. The vector is used to weight different feature embeddings via element-wise product, i.e., $\mathbf{V} =$

$[\mathbf{v}_1, \cdots, \mathbf{v}_N] = [\mathbf{a}_1\mathbf{e}_1, \cdots, \mathbf{a}_N\mathbf{e}_f]$ (V is the weighted embedding matrix). In this way, features are promoted or demoted based on their importance in different domains, which enables the model to adaptively find proper feature sets to enhance domain modeling.

2.3 Domain-adaptive Feature Representation

After feature selection, the model learns hidden feature representations by modeling feature relatedness [18]. Since features may form various relations in different domains, we use a domain-adaptive feature representation module with K stacked adaptive blocks to generate domain-aware feature representations. Taking the first block as an example, we flatten $\mathbf{V}$ into a vector $\mathbf{v}$ and transform it by $\mathbf{h} = \tanh(\mathbf{W}_h\mathbf{v} + \mathbf{b}_h)$, where $\mathbf{W}_h$ and $\mathbf{b}_h$ are parameters. This vector is further processed by M multi-layer perceptrons (MLPs) to generate multiple feature representations, where a domain-adaptive router picks the most suitable one as domain-aware feature representations (we prefer top-1 selection due to its inference efficiency). The raw weighting score s_i of the i-th MLP is as follows:

$$s_i = \frac{e^{\mathbf{w}_i^T \mathbf{e}_d}}{\sum_j e^{\mathbf{w}_j^T \mathbf{e}_d}}, \tag{3}$$

where $\mathbf{w}_i$ represent parameters. Since the top-1 selection operation is not differentiable, we use Gumbel-Softmax [7] to help back-propagation in model training. The final weighting score p_i of the i-th model is formulated as follows:

$$p_i = \frac{e^{(\frac{\log(s_i)+g_i}{\tau})}}{\sum_j e^{(\frac{\log(s_j)+g_j}{\tau})}}, \ g_i = -\log(-\log(u_i)), \ u_i \sim U(0,1). \tag{4}$$

Denote the output of the i-th MLP as $\mathbf{r}_i$, and the final output feature representation $\mathbf{r}$ for model training is computed by $\mathbf{r} = \sum_{i=1}^{M} p_i\mathbf{r}_i$. At the inference stage, only the MLP with the largest weighting score is activated, i.e., $\mathbf{r} = \mathbf{r}_i$, where $i = \text{argmax}(p_i)$. In this way, the model can select different submodules to process data in different domains without increasing computational costs in online inference. To facilitate gradient propagation, we introduce a residual connection between the input and output of each block. The final output from the top adaptive block is denoted as $\mathbf{r}'$.

2.4 Click Prediction

Finally, we compute the click probability score based on the feature representation $\mathbf{r}'$. We apply a linear layer to $\mathbf{r}'$ to get a click score $\hat{y}_r$ that reflects user-item relevance, which is formulated as $\hat{y}_r = \mathbf{w}_r^T\mathbf{r}' + b_r$ ($\mathbf{w}_r$ and b_r are parameters). To consider domain impacts on click prediction, we derive a domain bias score $\hat{y}_b$ from the domain embedding by $\hat{y}_b = \mathbf{w}_b^T\mathbf{e}_d + b_b$ ($\mathbf{w}_b$ and b_b are parameters). Both scores are added together to form a unified click probability score $\hat{y}$, which is further normalized by the sigmoid function. The loss function $\mathcal{L}$ for model training is the cross-entropy between the predicted click score and label on each instance.

3 EXPERIMENTS

3.1 Datasets

We conduct extensive experiments on two datasets. The first one is Ali-CCP [16][1], which is a public dataset gathered from real logs in Alimama. It contains 3 domains, 17 feature fields, and 8.5 million samples, where the training/test sets are split by time. The second one is a proprietary dataset (denoted as "Industrial") collected from our advertising platform in Mar. 2022. It contains 6 domains, 31 feature fields, and 2.2 million samples. We use the data in the first 3 days for training and validation, and use the rest data for test. Detailed statistics of the two datasets are summarized in Table 1. We can see that different domains have quite different CTRs, which is a strong indication of domain gaps.

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=408

Table 1: The example percentage and average CTR of each domain in Ali-CCP and Industrial.

Domains	Ali-CCP			Industrial					
	#1	#2	#3	#1	#2	#3	#4	#5	#6
Percentage	37.5%	61.7%	0.80%	80.4%	8.68%	3.84%	3.76	2.52%	0.77%
CTR	4.00%	3.81%	4.39%	5.03%	1.92%	8.23%	2.06%	27.5%	0.82%

3.2 Experimental Settings

Following common practices, we report the AUC of each domain and the overall AUC calculated on all samples as the performance metrics. In our experiments, the dimensions of feature embeddings are set to 5. We use 4 meta-embeddings for each feature, and use 3 adaptive blocks with 3 submodels in each block. We use Adam [10] for optimization and set the learning rate to $2e-5$. The value of τ is set to 1. The batch size is 128. The baselines are implemented based on the FuxiCTR library [28].

3.3 Offline Evaluation

We compare *AdaptiveCTR* with several baselines to verify its effectiveness. Two of them are single-domain methods, including (1) **DNN**, a recommendation model based on deep neural networks; and (2) **DeepFM** [4], a popular recommendation method based on deep factorization machine. In these methods, we mix all samples from different domains to train a unified model. Another baseline is **MMoE** [16], which is a canonical method for multi-task learning. Here we adapt it for multi-domain CTR prediction, where each fully-connected network is treated as an expert, and the number of experts is equal to the number of domains. The rest two baselines are multi-domain methods, including (1) **DADNN** [6], which employs a shared bottom model to learn generic feature representations and domain-specific layers with a domain router to generate the output for each domain; and **STAR** [19], the star topology fully-connected neural network which factorizes the model for each domain into a shared centered model and a domain-specific model. The results of these methods on the two datasets are shown in Table 2, from which we have several observations. First, compared with single-domain methods such as DNN and DeepFM, multi-task or multi-domain methods achieve better performance. This is because different domains may have huge gaps and it is usually suboptimal to simply mix multi-domain training data. In addition, we find DADNN has a similar performance as MMoE, which may be because the router mechanism in DADNN is similar to the mixture-of-expert mechanism in MMoE. STAR outperforms DADNN and MMoE, which is probably because it has more domain-specific model components and thereby can better overcome domain barriers. Moreover, AdaptiveCTR shows superior performance across most domains compared to other methods on both Ali-CCP and Industrial datasets

Table 2: Evaluation results of different methods on the Ali-CCP and Industrial datasets. The best results are in bold and the second best results are underlined for each column. The best overall AUC results are marked with ⋆.

Methods	Ali-CCP				Industrial						
	#1	#2	#3	Overall	#1	#2	#3	#4	#5	#6	Overall
DNN	0.5866	0.5838	0.5684	0.5849	0.8688	0.8435	0.8576	0.8490	0.8244	0.6554	0.8560
DeepFM	0.5989	0.5972	0.5773	0.5980	0.8693	0.8415	0.8548	0.8431	0.8247	0.6283	0.8566
MMoE	0.6046	0.6000	0.5825	0.6019	0.8703	0.8464	0.8581	0.8499	0.8251	0.6597	0.8574
DADNN	0.6041	0.5994	0.5836	0.6015	0.8700	0.8447	0.8589	0.8528	0.8219	0.6611	0.8577
STAR	0.6074	0.6021	0.5847	0.6036	0.8711	0.8516	0.8631	0.8523	0.8241	**0.6828**	0.8596
AdaptiveCTR	**0.6086**	**0.6117**	**0.5897**	**0.6102**⋆	**0.8729**	**0.8526**	**0.8655**	**0.8566**	**0.8270**	0.6725	**0.8621**⋆
%Improv.	0.20%	1.59%	0.86%	1.09%	0.21%	0.12%	0.28%	0.45%	0.23%	-1.51%	0.29%

Table 3: Online A/B test results on our advertising platform.

Domains	#1	#2	#3	#4	#5	#6
CVR	+4.3%	+18%	+3.1%	+15%	+4.5%	+12%
COPC	+1.6%	+8.4%	+3.4%	+3.4%	+10%	+21%

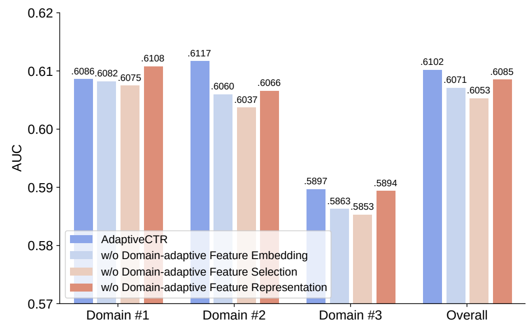

Figure 3: Ablation study of AdaptiveCTR.

(except for a minor *Domain#6* on Industrial). This is because our method can consider domain characteristics in all steps of our model, thereby is less suffered from domain gaps due to its high domain adaptivity. In addition, since some model components (e.g., meta-embeddings and MLP models) are aware of information in all domains, the commonalities between domains can also be effectively encoded. Thus, AdaptiveCTR is more effective than baselines in multi-domain CTR prediction.

3.4 Online Evaluation

We deploy AdaptiveCTR online and conduct an A/B test on a real-world CVR (post-click conversion rate) recommendation scenario in our online advertising platform. We use CVR and conversion over predicted conversion (COPC) as the metrics. Table 3 shows the conversion rate improvements over a well-performed single-domain model from April 15th to May 15th. Note that the baseline models are trained separately on the logs of each domain. From the results, we observe notable CVR and COPC improvements over the base model in all domains. For example, we achieve 4.3% of CVR improvement and 1.6% of COPC improvement in Domain#1, which is a major domain that occupies around 80% of clicks. This shows that the data from minor domains can provide complementary information to improve recommendation performance in major domains. In the rest domains, the average improvements are 17.4% and 8.5% in terms of CVR and COPC. It reveals that the rich supervision signals in multi-domain data can alleviate the data sparsity problem and help boost the model performance. These results demonstrate the effectiveness of our approach in improving the model recommendation performance in both major and minor domains. Up to now, our method has been deployed on our online advertising platform to serve multiple-domain traffic with a single model.

3.5 Ablation Study

Here we verify the effectiveness of different components in AdaptiveCTR. The AUC scores of AdaptiveCTR without domain-adaptive feature embedding, domain-adaptive feature selection, or domain-adaptive feature representation on the Ali-CCP dataset are shown in Fig. 3 (the results on the Industrial dataset show similar patterns). We find that domain-adaptive feature selection plays the most important role in our method. It reflects that selecting salient features in a domain-aware way is important for overcoming domain gaps. In addition, the domain-adaptive feature embedding mechanism also contributes to the improvements. This shows that adaptively learning domain-specific embeddings for the same feature can help exploit multi-domain information. Besides, although AdaptiveCTR achieves the best performance in Domain#1 if the feature representation module is not domain-adaptive, its overall performance is suboptimal. It is because the domain-adaptive feature representation module can help learn more domain-discriminative feature representations. In summary, the ablation studies show the contribution of each component in our method.

4 CONCLUSION

In this paper, we propose a multi-level domain adaptation method named AdaptiveCTR for multi-domain CTR prediction. It introduces domain adaptability to the main steps of CTR prediction, including feature embedding, feature selection, and feature representation. By using domain information to guide these processes, the model can fully overcome the huge barriers between domains and automatically learn informative features according to domain characteristics. Extensive offline experimental results on a public dataset and an internal dataset show the effectiveness of our AdaptiveCTR method. Furthermore, a one-month online A/B test on our advertising platform validates the superiority of AdaptiveCTR in online environments. AdaptiveCTR has become a major model for multiple-domain Ad traffic serving in our system.

REFERENCES

[1] Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. 2019. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In *AAAI*, Vol. 33. 94–101.

[2] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. 2019. Cross-domain recommendation without sharing user-relevant data. In *WWW*. 491–502.

[3] Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. 2021. An embedding learning framework for numerical features in ctr prediction. In *KDD*. 2910–2918.

[4] 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.

[5] Xiaobo Hao, Yudan Liu, Ruobing Xie, Kaikai Ge, Linyao Tang, Xu Zhang, and Leyu Lin. 2021. Adversarial Feature Translation for Multi-domain Recommendation. In *KDD*. 2964–2973.

[6] Junyou He, Guibao Mei, Feng Xing, Xiaorui Yang, Yongjun Bao, and Weipeng Yan. 2020. DADNN: Multi-Scene CTR Prediction via Domain-Aware Deep Neural Network. *arXiv preprint arXiv:2011.11938* (2020).

[7] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparametrization with Gumble-Softmax. In *ICLR*.

[8] Yuchen Jiang, Qi Li, Han Zhu, Jinbei Yu, Jin Li, Ziru Xu, Huihui Dong, and Bo Zheng. 2022. Adaptive Domain Interest Network for Multi-domain Recommendation. In *CIKM*. 3212–3221.

[9] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-supervised learning for cross-domain recommendation to cold-start users. In *CIKM*. 1563–1572.

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

[11] Pan Li and Alexander Tuzhilin. 2020. Ddtcdr: Deep dual transfer cross domain recommendation. In *WSDM*. 331–339.

[12] Bin Liu, Niannan Xue, Huifeng Guo, Ruiming Tang, Stefanos Zafeiriou, Xiuqiang He, and Zhenguo Li. 2020. AutoGroup: Automatic feature grouping for modelling explicit high-order feature interactions in CTR prediction. In *SIGIR*. 199–208.

[13] 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.

[14] Weiming Liu, Xiaolin Zheng, Mengling Hu, and Chaochao Chen. 2022. Collaborative Filtering with Attribution Alignment for Review-based Non-overlapped Cross Domain Recommendation. In *WWW*. 1181–1190.

[15] Linhao Luo, Yumeng Li, Buyu Gao, Shuai Tang, Sinan Wang, Jiancheng Li, Tanchao Zhu, Jiancai Liu, Zhao Li, Binqiang Zhao, et al. 2022. MAMDR: A Model Agnostic Learning Method for Multi-Domain Recommendation. *arXiv preprint arXiv:2202.12524* (2022).

[16] 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*. 1930–1939.

[17] Weizhi Ma, Min Zhang, Chenyang Wang, Cheng Luo, Yiqun Liu, and Shaoping Ma. 2018. Your Tweets Reveal What You Like: Introducing Cross-media Content Information into Multi-domain Recommendation.. In *IJCAI*. 3484–3490.

[18] Wentao Ouyang, Xiuwu Zhang, Shukui Ren, Chao Qi, Zhaojie Liu, and Yanlong Du. 2019. Representation learning-assisted click-through rate prediction. In *IJCAI*. 4561–4567.

[19] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, et al. 2021. One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction. In *CIKM*. 4104–4113.

[20] Yu Zhang, Bin Cao, and Dit-Yan Yeung. 2010. Multi-domain collaborative filtering. In *UAI*. 725–732.

[21] Zihan Zhang, Xiaoming Jin, Lianghao Li, Guiguang Ding, and Qiang Yang. 2016. Multi-domain active learning for recommendation. In *AAAI*.

[22] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-domain recommendation via preference propagation graphnet. In *CIKM*. 2165–2168.

[23] 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 *AAAI*, Vol. 33. 5941–5948.

[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 *KDD*. 1059–1068.

[25] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. Dtcdr: A framework for dual-target cross-domain recommendation. In *CIKM*. 1533–1542.

[26] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-domain recommendation: challenges, progress, and prospects. In *IJCAI*. 4721–4728.

[27] Jieming Zhu, Qinglin Jia, Guohao Cai, Quanyu Dai, Jingjie Li, Zhenhua Dong, Ruiming Tang, and Rui Zhang. 2023. FINAL: Factorized Interaction Layer for CTR Prediction. In *SIGIR3*. 2006–2010.

[28] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open Benchmarking for Click-Through Rate Prediction. In *CIKM*. 2759–2769.

[29] Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021. Transfer-meta framework for cross-domain recommendation to cold-start users. In *SIGIR*. 1813–1817.

[30] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized transfer of user preferences for cross-domain recommendation. In *WSDM*. 1507–1515.