DFFM: Domain Facilitated Feature Modeling for CTR Prediction

Wei Guo*
guowei67@huawei.com
Huawei Noah's Ark Lab
Singapore

Chenxu Zhu*
zhuchenxu1@huawei.com
Huawei Noah's Ark Lab
China

Fan Yan*
yanfan6@huawei.com
Huawei Noah's Ark Lab
China

Bo Chen
chenbo116@huawei.com
Huawei Noah's Ark Lab
China

Weiwen Liu
liuweiwen8@huawei.com
Huawei Noah's Ark Lab
China

Huifeng Guo
huifeng.guo@huawei.com
Huawei Noah's Ark Lab
China

Hongkun Zheng
zhenghongkun1@huawei.com
Huawei Technologies Co Ltd
China

Yong Liu
liu.yong6@huawei.com
Huawei Noah's Ark Lab
Singapore

Ruiming Tang
tangruiming@huawei.com
Huawei Noah's Ark Lab
China

ABSTRACT

Recently, numerous models have been proposed that attempt to use a unified model to serve multiple domains. Although much progress has been made, we argue that they ignore the importance of feature interactions and user behaviors when modeling cross-domain relations, which is a coarse-grained utilizing of domain information. To solve this problem, we propose Domain Facilitated Feature Modeling (DFFM) for CTR prediction. It incorporates domain-related information into the parameters of the feature interaction and user behavior modules, allowing for domain-specific learning of these two aspects. Extensive experiments are conducted on two public datasets and one industrial dataset to demonstrate the effectiveness of DFFM. We deploy the DFFM model in Huawei advertising platform and gain a 4.13% improvement of revenue on a two week online A/B test. Currently DFFM model has been used as the main traffic model, serving for hundreds of millions of people.

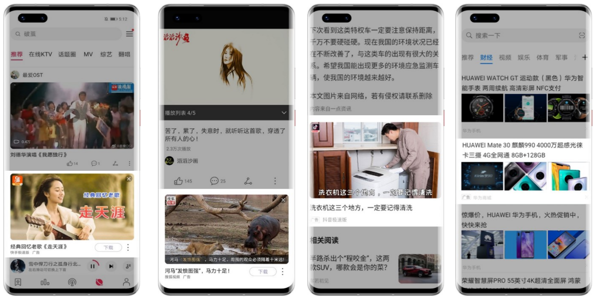
CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

Click-through Rate, Multi-domain, Feature Modeling

ACM Reference Format:
Wei Guo, Chenxu Zhu, Fan Yan, Bo Chen, Weiwen Liu, Huifeng Guo, Hongkun Zheng, Yong Liu, and Ruiming Tang. 2023. DFFM: Domain Facilitated Feature Modeling for CTR Prediction . In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583780.3615469



(a) Music (b) Video (c) Reading (d) News

Figure 1: Four representative business domains of Huawei advertising platform.

1 INTRODUCTION

Click-through rate (CTR) prediction is crucial in industrial recommender systems. They can be divided into two categories: feature interaction modeling based and user behavior feature mining based [28]. For the former methods, such as DeepFM [8], AutoInt [20] and FibiNet [13], attempt to construct explicit feature interactions to memory feature correlations better. For the latter direction, representative models are DIN [31], DIEN [30] and CAN [29]. They try to compute the attention weights of different historical items based on the target item, to get a more comprehensive understanding of user interests.

In real-world web applications, there are usually a number of business domains for recommendations. Figure 1 shows some representative domains on the Huawei advertising platform. It can be seen that different domains have different display styles, such as position, size, layout and so on, which leads to domain distribution discrepancy problem. Early industrial practices train separate models for each domain with data from each domain, or train a unified model by mixing data from all domains [19, 32]. However, separate models have the problem of low data volume, since many domains have a small amount of data, and results in high labor costs, as

[*] Co-first authors with equal contributions.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615469

there are too many domains to maintain. The unified model may not work well in all domains, and may lead to large prediction biases due to the different domain distribution.

To solve the distribution discrepancy problem, multi-domain CTR prediction has been proposed [19], which aims to learn an effective and robust model that can adapt to different domains simultaneously. Most of the existing methods proposed for multi-domain CTR prediction can be classified into two categories: multi-tower models and single-tower models. Multi-tower models like Shared Bottom [4], MMOE [16], PLE [22] and STAR [19] are adopted from the multi-task learning (MTL) [5], in which each tower is used to predict a specific domain. Models of the latter class such as AdaSparse [27] and APG [26] build single tower networks with different parameters generated based on domain information to tackle the multi-domain problem. Although these two types of multi-domain CTR prediction methods have made great progress by accounting for domain differences, two challenges remain:

- **Domain Discrepancy of Feature Interaction.** Feature interactions are useful for identifying the actual reasons for user's actions. For example, price and brand are key factors for shopping domain, while category and vocalist are more significant in the music domain, indicating that the 2-order interaction patterns have different importance in different domains. Multi-tower models use a limited number of experts to extract information, neglecting the fine-grained analysis of feature interactions. Single-tower models alleviate the problem of domain adaption from the perspective of network architecture for each domain, but lacks insight into differences between feature interactions.
- **Domain Discrepancy of User Behavior.** Models based on user behavior features mining capture user's real interest by understanding user's past interactions. However, user behaviors differs greatly among different domains. For example, people may be broadly interested in music while concentrating on games. If we treat all interactions equally without distinguishing which domain they come from, we might recommend items related to interests from other domains to users on this domain, which would inevitably reduce the accuracy of the recommendation.

To address these problem, we propose a **D**omain **F**acilitated **F**eature **M**odeling (DFFM) framework for multi-domain recommendation. To tackle the first challenge, we employ a domain-specific feature transformation for each feature, and then use it for feature interaction. To model the feature transformation, inspired by CAN [29], we project the domain feature embedding onto parameters of a deep neural network, then feed the original features to this network to obtain domain-specific representations.

In conclusion, our contributions can be summarized as follows:

- We emphasize the importance of domain-aware feature interaction and user behavior modeling in multi-domain CTR prediction. To our knowledge, this is the first work that attempts to solve the multi-domain problem from the perspective of these two issues.
- We propose the framework DFFM, to integrate domain-related information into the parameters of the feature interaction and user behavior modules. DFFM is able to improve the performance of existing CTR models in the above two aspects from a multi-domain perspective.
- We conduct experiments on public and industrial datasets compared with different baseline models to demonstrate the superiority of DFFM. Online results further confirm the effectiveness and applicability of DFFM.

2 RELATED WORK

Single-Domain CTR prediction. To depict the correlations among features, effectively modeling of feature interaction is the key of accurate CTR prediction [9–11, 15, 33]. Deep & Cross Network (DCN) [23] applies layer-wise feature crossing recursively, thus feature interactions at different orders can be captured. DCN V2 [25] further improves DCN by upgrading the feature crossing vector to a matrix to enhance its representing ability. AutoInt [20] is proposed to utilize the self-attention architecture to learn feature interactions, which obtains not only superior performance but also good interpretability. Besides feature interaction modeling, user behavior feature modeling is also very important, which has attracted a lot of research interests. DIEN [30] utilizes both GRU units and target attention to model interest evolution process and extract target related interests. CAN [29] enhances the interaction between target item and historical items by designing a co-action network with dynamic network parameter generation. Despite the success of the above methods, none of them can effectively handle the severe distribution differences when samples of multiple domains are used. To solve the multi-domain problem, a number of multi-domain modeling methods are proposed.

Multi-Domain CTR prediction. The distribution of data among domains is usually very different. To solve this problem, the Multi-Task Learning (MTL) paradigm [5] which learns the shared and specific experts firstly and then applying adaptive gates to select relevant information for prediction is widely used. Representative models includes Shared Bottom [4], MMOE [16] and PLE [22]. Shared Bottom combines a shared bottom layer with multiple prediction layers for multi-domain modeling. MMOE introduces different gating networks to obtain different weights over experts for prediction. PLE further proposes shared and specific expert network to mitigate the task conflicts. Recently, with the development of domain adaptation (DA) [2, 3], more customized multi-domain modeling methods have emerged. STAR [19] proposes a star topology structure with partitioned normalization to handle the domain distribution discrepancy. AdaSparse [27] introduces neuron-level weight pruning to learn domain adaptive network architecture for different domains. Although the sample distribution discrepancy is taken into account by these multi-domain methods, none of them attempt to solve the issue from the viewpoint of feature interaction and user behavior feature modeling, which limits their performance.

3 PRELIMINARY

The CTR dataset can be denoted as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, where x_i and $y_i \in \{0, 1\}$ represent the feature set and label of the i-th sample, respectively. In real-world recommendation, CTR prediction models have to deal with multiple different domains. The domain segmentation is usually based on some context features such as *scenario_id* or *position*, but can be extended to some features such as *user_profile*,

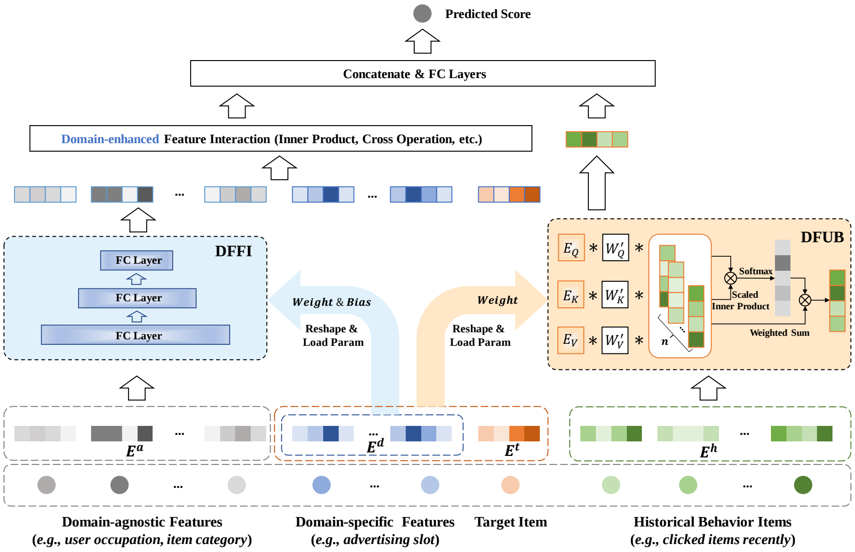


Figure 2: The overall framework of DFFM.

item_category, etc. Then the dataset $\mathcal{D}$ will be partitioned into mutiple domain-specific subsets (i.e., $\mathcal{D} = \mathcal{D}^1 \bigcup \cdots \bigcup \mathcal{D}^M$), where the m-th subset $\mathcal{D}^m = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}^m|}$ is obtained according to the specific domain. The goal of multi-domain CTR prediction is to learn a unified model to perform well on all domains.

Specifically, the whole feature set $\boldsymbol{x}$ can be divided into four part: domain-agnostic feature set $\boldsymbol{x}_a$, domain-specific feature set $\boldsymbol{x}_d$, target item feature set $\boldsymbol{x}_t$ and historical item feature set $\boldsymbol{x}_h$. All these four feature are first transformed to the embedding layer as $\mathbf{E} = \{\mathbf{E}^a, \mathbf{E}^d, \mathbf{E}^t, \mathbf{E}^h\} = \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n\}$, where n is the number of fields. Then we try to predict the CTR for different domains as:

$$\hat{y} = f\left(\boldsymbol{x}_a, \boldsymbol{x}_d, \boldsymbol{x}_t, \boldsymbol{x}_h; \theta\right) , \tag{1}$$

where $f(\cdot)$ is the learned function with parameter θ.

4 METHOD

4.1 Overall Framework

The overall framework of Domain Facilitated Feature Modeling (DFFM) can be divided into two major modules: Domain Facilitated Feature Interaction (DFFI) and Domain Facilitated User Behavior (DFUB). The overview of our proposed DFFM is shown in Figure 2.

The DFFI module is responsible for bringing valuable domain information to support the feature interaction modeling while the DFUB module takes advantage of both domain and target knowledge to better modeling the user behaviors. The detailed design of these two modules will be illustrated in next subsections.

4.2 DFFI

Feature interaction modeling is essential in CTR prediction, such as PNN, AutoInt and DCN. However, all these models ignore the importance of considering domain information. Feature interactions should be modeled differently in various domains, e.g., some feature interactions are extremely important in some domains while meaningless in some other domains. Therefore, we propose Domain Facilitated Feature Interaction (DFFI) module, which can act on any feature interaction model with introducing domain information to assist the interaction modeling.

Take the inner product in PNN [18] as an example, our domain-enhanced inner product can be formulated as:

$$\begin{aligned} \mathcal{F}_{\text{domain}}\left(\boldsymbol{e}_i, \boldsymbol{e}_j\right) &= \left\langle \boldsymbol{e}_i, \boldsymbol{e}_j \right\rangle_{\text{domain}} , \\ &= \left\langle \mathcal{D}\left(\boldsymbol{e}_i\right), \mathcal{D}(\boldsymbol{e}_j) \right\rangle , \end{aligned} \tag{2}$$

where $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ are the embedding of the i-th and j-th field; $\langle \cdot, \cdot \rangle$ is the inner product; $\mathcal{D}\left(\cdot\right)$ is the domain network to fuse the domain information, which is a micro-MLP and can be formulated as:

$$\mathbf{h}^{(0)} = \mathbf{h}_{input} , \tag{3}$$

$$\mathbf{h}^{(k)} = \sigma(\mathbf{W}^{(\mathbf{k-1})}\mathbf{h}^{(k-1)} + \mathbf{b}^{(\mathbf{k-1})}), \quad k \in [1, \cdots, L] , \tag{4}$$

$$\mathcal{D}(\mathbf{h}_{input}) = \mathbf{h}^{(L)} , \tag{5}$$

where $\mathbf{h}_{input}$ is the input vector, σ is the activation function, L is the depth of the domain network. In particular, the weight and bias parameters $\mathbf{W}^{(\mathbf{k})}$ and $\mathbf{b}^{(\mathbf{k})}$ are generated by the concatenate domain embedding by reshaping and splitting. Formally:

$$\mathbf{W}^{(\mathbf{k})}, \mathbf{b}^{(\mathbf{k})} = Reshape(Split(\mathbf{E}^d)) , \tag{6}$$

where $\mathbf{E}^d$ is the concatenation of all domain embedding. A visual illustration of this process is shown in the Figure 2. It should be noted that PNN only models 2-order feature interaction. For models with high-order feature interactions like DCN and AutoInt, we should apply the domain network to the input of each order feature interaction.

With this domain-enhanced inner product, then we can achieve the domain-enhance feature interaction layer as

$$\mathbf{h}_{\text{domain}} = Concat\left(\mathcal{F}_{\text{domain}}\left(\boldsymbol{e}_1, \boldsymbol{e}_2\right), \cdots, \mathcal{F}_{\text{domain}}\left(\boldsymbol{e}_n, \boldsymbol{e}_{n-1}\right)\right) \tag{7}$$

$$\mathbf{h}_{DFFI} = MLP\left(Concat\left(\mathbf{h}_{\text{domain}}, \mathbf{E}\right)\right) , \tag{8}$$

where $\mathbf{h}_{DFFI}$ is the output representation of DFFI module; n is the number of fields; $\mathbf{E}$ is the concatenate embedding of all fields.

In this way, the modeling of feature interactions takes into account the domain knowledge through dynamic weighted network. The same approach can work not only on inner products but also for other feature interaction models, such as self-attention mechanism in AutoInt or cross network in DCN, which will be verified in Section 5.4.2.

4.3 DFUB

User behavior modeling plays an important role of CTR prediction [12]. Current approaches like DIN and CAN focus on modeling the co-action between target items and user historical sequences. Despite the success of these approach in specific domains, we emphasize the importance of considering domain related information of modeling user behavior. For example, users' buying interests vary rapidly between pre-sales and post-sales domains, which indicates different co-action patterns of their behavior history on the click action. To explicitly consider this phenomenon, we propose Domain Facilitated User Behavior modeling, abbreviated as DFUB.

DFUB takes modified multi-head self-attention mechanism to process user history sequence. Assume the embedding of user behavior history can be denoted as $\mathbf{E}^h = \{\boldsymbol{e}_1^h, \boldsymbol{e}_2^h, \cdots, \boldsymbol{e}_n^h\}$, where n is the behavior history length. The target item and domain features denoted as $\mathbf{E}^t = \{\boldsymbol{e}^t\}$ and $\mathbf{E}^d = \{\boldsymbol{e}^d\}$. For the i-th head, the standard self-attention module process can be formulated as Equation 9.

$$\mathbf{head}_i = SelfAttn^{(i)}_{\Theta(\mathbf{E}^t,\mathbf{E}^d)}(\mathbf{E}^h) = Softmax(\frac{\mathbf{Q}_i\mathbf{K}_i^{\top}}{\sqrt{d_i}})\mathbf{V}_i \ . \tag{9}$$

In Equation 9, d_i is the output dimension of $\mathbf{head}_i$, and $\Theta(\mathbf{E}^t, \mathbf{E}^d)$ means using $\mathbf{E}^t, \mathbf{E}^d$ to get the parameter of the self-attention module. $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i$ refer to the query matrix, key matrix, and value matrix. The three matrix are all calculated from the behavior sequential embedding $\mathbf{E}^{\mathbf{h}}$ as Equation 10.

$$\mathbf{Q}_i = \mathbf{E}^h\mathbf{W}_Q^{(i)}, \quad \mathbf{K}_i = \mathbf{E}^h\mathbf{W}_K^{(i)}, \quad \mathbf{V}_i = \mathbf{E}^h\mathbf{W}_V^{(i)} \ . \tag{10}$$

The $\mathbf{W}_Q^{(i)}, \mathbf{W}_K^{(i)}, \mathbf{W}_V^{(i)}$ are transformation matrix. So far the presented calculation process is all about internal item interaction in the sequential behavior items. The goal is to fully interact the target item and historical behavior items, and utilize the domain information to guide this interaction. We try to achieve this by integrate the target embedding $\mathbf{E}^{\mathbf{t}}$ and the domain embedding $\mathbf{E}^{\mathbf{d}}$ into self-attention parameters directly according to Equation 11.

$$\begin{aligned}
\mathbf{W}_Q^{(i)} &= \mathbf{W}_Q^{(i)\prime}\mathbf{E}_Q^{\top}, \quad \mathbf{E}_Q = Reshape(\mathbf{E}^t, \mathbf{E}^d) \\
\mathbf{W}_K^{(i)} &= \mathbf{W}_K^{(i)\prime}\mathbf{E}_K^{\top}, \quad \mathbf{E}_K = Reshape(\mathbf{E}^t, \mathbf{E}^d) \\
\mathbf{W}_V^{(i)} &= \mathbf{W}_V^{(i)\prime}\mathbf{E}_V^{\top}, \quad \mathbf{E}_V = Reshape(\mathbf{E}^t, \mathbf{E}^d)
\end{aligned} \tag{11}$$

As shown in Equation 11, the parameter of self-attention can be decomposed into two parts. The first part $(\mathbf{W}_Q^{(i)\prime}, \mathbf{W}_K^{(i)\prime}, \mathbf{W}_V^{(i)\prime})$ is the inherent parameter of the self-attention module. The second part $(\mathbf{E}_Q, \mathbf{E}_K, \mathbf{E}_V)$ is reshaped from the embedding of target item and domain related features. In this way the information from target item and domain features can get involved into the interaction process element-wisely and thoroughly. Multiple head concatenation makes up an integrated multi-head self-attention layer, which can be concatenated to make up the final output of DFUB module as Equation 12, in which the $\mathbf{W}^{out}$ refers to the output transformation matrix:

$$\mathbf{h}_{DFUB} = Concat(\mathbf{head}_1, ..., \mathbf{head}_n)\mathbf{W}^{out} \ . \tag{12}$$

4.4 Prediction Layer and Optimization

Then we can concatenate the output representation from DFFI and DFUB module together to obtain

$$\mathbf{h}_{output} = Concat\left(\mathbf{h}_{DFFI}, \mathbf{h}_{DFUB}\right) \ . \tag{13}$$

A linear layer with a Sigmoid function is used to obtain the final prediction:

$$\hat{y} = Sigmoid(\mathbf{W}_o\mathbf{h}_{output} + \mathbf{b}_o) \ , \tag{14}$$

where $\mathbf{W}_o$ and $\mathbf{b}_o$ are the weight and bias parameters. We adopt the widely used cross entropy loss as the objective function.

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Datasets. We assess the efficacy of our proposed approach on two large-scale public datasets (*Ali-CCP* and *Ali-Mama*) and an industry dataset. Table 1 shows the statistics of the public datasets and the industry dataset is described in Section 5.3.

Table 1: Dataset statistics.

Dataset	Users	Items	Feat.	Beha.	Domains
Ali-CCP	265,945	1,010,133	18	4	3
Ali-Mama	793,494	468,920	14	2	3

- **Ali-CCP**[1]: This is a dataset collected from real-world user behavior records of Taobao [17]. It contains two parts: training logs and testing logs. We divide the training logs into training set and validation set at random in a 9:1 ratio. Each sample contains a variety of features: such as user gender, age, occupation, item category, shop and so on. We also employ user behavior features of category sequence, shop sequence, brand sequence and intention node sequence. We divided the dataset into 3 domains according to the *position_id.*
- **Alimama**[2]: This dataset is provided by Alimama [7], an online advertising platform in China. It is made up of 8 days of ad records from 2017. We take 90% of data from the first 7 days as training set, the rest as the validation set, and the last day as the testing set. The user profile comprises information such as group, age, occupation, etc. The item characteristics include things like category, brand, customer and so on. Two user behavior features are used: category list and brand list. We divided the dataset into 3 domains based on the feature *pvalue_level_id.*

[1] https://tianchi.aliyun.com/dataset/408

[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

Table 2: The overall performance on Ali-CCP and Ali-Mama datasets in terms of AUC. Boldface denotes the highest score and underline indicates the best result of the baselines. $\star$ represents significance level p-value < 0.05.

Dataset		Ali-CCP				Ali-Mama			
Group	Model	All	S1	S2	S3	All	S1	S2	S3
Base	DNN	0.5678	0.5689	0.5636	0.5664	0.5638	0.5585	0.5662	0.5635
FI	PNN	0.5700	0.5718	0.5648	0.5683	0.5644	0.5591	0.5661	0.5665
	AutoInt	0.5671	0.5681	0.5637	0.5657	0.5646	0.5600	0.5665	0.5649
	DCN V2	0.5771	0.5788	0.5663	0.5753	0.5651	0.5595	0.5669	0.5672
	FiBiNET	0.5706	0.5718	0.5613	0.5691	0.5642	0.5576	0.5673	0.5627
MDM	Shared Bottom	0.5864	0.5909	0.5715	0.584	0.5648	0.5596	0.5675	0.5652
	MMOE	0.5867	0.5906	0.5664	0.5843	0.5654	0.5581	0.568	0.5678
	PLE	0.5893	0.5937	0.5714	0.5864	0.5675	0.5635	0.5698	0.5679
	STAR	0.5895	0.5932	0.5702	0.5877	0.5666	0.5729	0.5716	0.5644
	APG	0.5873	0.5917	0.5628	0.5841	0.5714	0.5665	0.5727	0.5741
	AdaSparse	0.5887	0.5926	0.574	0.5859	0.5680	0.5625	0.5708	0.5661
User Hisotry	DIN	0.5909	0.5955	0.5732	0.5875	0.5715	0.5696	0.5727	0.5698
	DIEN	0.5903	0.5937	0.5737	0.5877	0.5727	0.5692	0.5739	0.5732
	CAN	0.5914	0.5964	0.5740	0.5878	0.5729	0.5724	0.5721	0.5758
Our	DFFM	**0.5969***	**0.5990***	**0.5771***	**0.5951***	**0.5848***	**0.5820***	**0.5867***	**0.5850***

5.1.2 Baselines and Evaluation Metric. To demonstrate the effectiveness of our proposed model, we compare DFFM to three classes of existing models. (A) feature interaction based models(PNN [18], AutoInt [21], DCN V2 [25], FiBiNET [13]); (B) user behavior feature based models(DIN [31], DIEN [30], CAN [29]) (C)Multi-Domain models (Shared Bottom [4], MMOE [16], PLE [22],STAR [19] APG [26] and AdaSparse [27]). We use AUC as the evaluation metric. A higher AUC score indicates a better performance.

5.1.3 Parameter Settings. We use Adam [14] optimizer to optimize different models. For fair comparison, we fix the embedding size as 16, the batch size as 1024 and the deep layers as {128, 128, 128} for all models. The learning rate is searched from {1e-1,1e-2,1e-3,1e-4}. We tune L_2 regularization coefficient from {0, 1e-1, 1e-2, 1e-3, 1e-4}, and the dropout ratio from 0 to 0.5. The validation set is used to tune hyper-parameters, while the testing set yields the best results.

5.2 Experimental Results

In this section, we compare the performance of DFFM with the baseline models. Table 2 shows the experimental results of all compared models on two datasets. We have the following observations:

- **Both feature interactions and user behavior features bring benefits to model performance**. Despite the diverse model architectures, we can see a trend that feature interaction and user behavior feature based models perform significantly better than base models without these two aspects. For example, DCN V2 and CAN outperform DNN in terms of performance. However, special designs should be conducted to full exploit feature interaction and user behavior, since DCN V2 and CAN exceed PNN and DIN.
- **Multi-domain modeling improves performance**. There is a considerable performance gain when comparing multi-domain models to DNN and feature interaction models. This is related to multi-domain models' capacity to handle data distribution disagreement. Noticed that the improvement in Ali-Mama dataset is less significant than in Ali-CCP dataset. One possible reason is that the majority of these multi-domain baseline models employ DNN as the expert or backbone structure, which does not capture feature interaction adequately.
- **The superior performance of DFFM.** We can observe from Table 2 that DFFM consistently yields the best performance on all datasets. Concretely, DFFM beats the best baseline by **0.93%**, **0.44%**, **0.54%** and **1.24%** on Ali-CCP dataset (**2.08%**, **1.68%**, **2.55%** and **1.60%** on Ali-Mama dataset). This suggests that studying feature interaction and user behavior feature from the standpoint of domain adaptation can improve CTR prediction performance.

Table 3: The overall performance over industrial dataset in terms of AUC, gAUC and LogLoss.

Method Type	Method Name	AUC	gAUC	LogLoss
Feature Interaction	PNN [18]	0.8131	0.6720	0.3641
	DCN [24]	0.8111	0.6702	0.3673
	EDCN [6]	0.8131	0.6712	0.3653
	DeepFM [8]	0.8103	0.6688	0.3685
	AutoInt [20]	0.8131	0.6716	0.3657
	FiBiNET [13]	0.8119	0.6704	0.3668
User History	DIN [31]	0.8167	0.6755	0.3618
	CAN [29]	0.8168	0.6757	0.3627
	DIEN [30]	0.8172	0.6752	0.3642
Our	DFFM	**0.8186**	**0.6761**	**0.3571**

5.3 Industrial Experiments

5.3.1 Offline Industrial Experiments. We also test our approach on an offline industrial dataset, which is a million-scale dataset drawn from the click logs of the online ad platform and has more than 400 million impressions. We split them into training/dev/test sets by timestamp with 6:1:1 proportion.

It is worth noting that the domain related features in our industrial dataset, like site ID and task ID, contains thousands of values. As most multi-domain approaches contains specific model parameter for each domain, using these approaches will result in rapid expansion of the model size and insufficient parameter training. Besides, it is impossible to show thousands of domain results here. So we do not compare with multi-task or multi-domain approaches like MMoE and PLE. And only the overall performance is reported, in terms of AUC, gAUC, and LogLoss.

Table 4: Online A/B testing results of DFFI and DFFM compared with the base model M_{base}.

Metrics	M_{base}	M_{dffi}	M_{dffm}
eCPM	-	+1.77%	+4.13%
Latency	-	+3.40%	+16.0%

From Table 3, we summarize the observations in two points. Firstly, the user history modeling methods can consistently outperforms pure feature interaction models,which shows the importance of explicitly modeling of user history. Secondly, our DFFM outperforms all the other models, demonstrating the value of adding domain information for both feature interaction and user behavior modeling. Although it is not realistic to show the results of thousands of tiny domains, the metric gAUC provides a user-granularity perspective for fine-grained model performance, in which DFFM shows its advantage over other methods.

5.3.2 *Online Industrial A/B Test.*

To evaluate the performance of our approach in the real environment, we conduct online A/B test in our online advertising platform for consecutive two weeks, the last week of April and the first week of May in 2023. The compared baseline is denoted as M_{base}, which is a highly-optimized parallel-structure model. We deploy the feature interaction model DFFI denoted as M_{dffi} and the complete model DFFM with user behavior modeling denoted as M_{dffm}. Both M_{dffi} and M_{dffm} take the M_{base} as the base model to build up feature interaction and user behavior modeling blocks incrementally. Each model is trained over the latest click log, where an identical data process procedure is performed to ensure the comparability. We deploy each model on a single cluster. For online serving, all the three model are allocated 5% of the overall traffic. The deploy scenario contains hundreds of sites and mobile applications, where millions of daily active users interact with ads and tens of millions of user logs are generated every day. We compare the performance according to Effective Cost Per Mile(eCPM), which is widely used for online evaluation.

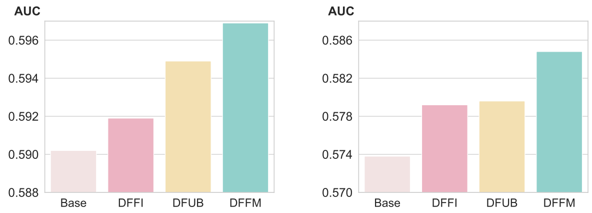
Table 4 shows the results of the three models, among which M_{dffm} performs the best and achieves 4.13% improvements with respect to eCPM. The M_{dffi} also ourperforms the baseline on eCPM for 1.77%. Moreover, the serving latency shown on the second row reveals that our approach brings significant performance gain with 16.0% inference time increased, which is acceptable. After 2 weeks of evaluation, the DFFM has become the main model in this scenario to carry most of the online traffic.

5.4 Ablation Study

5.4.1 *The Effectiveness of different modules in DFFM.*

We take FiBiNET+DIN as the base model and verify the superiority of DFFI and DFUB on this base model, respectively. The results are shown in Figure 3, from which we can draw the following conclusions.

- Comparing DFFI with the base model, we can observe that introducing domain information to feature interactions can significantly improve the prediction accuracy.
- Besides, the performance gap between DFUB and the base model indicates that model user behaviors with domain knowledge conduces to better performance, which is the functionality of DFUB module.
- DFFM achieves the best performance over these variants, substantiating that considering domain knowledge for feature interaction modeling and user behavior modeling can jointly boost the performance.



(a) Ali-CCP (b) Ali-Mama

Figure 3: Ablation study about different modules over Ali-CCP and Ali-Mama datasets in terms of AUC.

Table 5: Transferability analysis over Ali-CCP and Ali-Mama datasets in terms of AUC.

Datasets	Ali-CCP				Ali-Mama			
Doamin	All	D1	D2	D3	All	D1	D2	D3
PNN	0.5700	0.5718	0.5648	0.5683	0.5644	0.5591	0.5661	0.5665
+DFFI	**0.5775**	**0.5781**	**0.5688**	**0.5766**	**0.5693**	**0.5656**	**0.5706**	**0.5717**
AutoInt	0.5671	0.5681	0.5637	0.5657	0.5646	0.5600	0.5665	0.5649
+DFFI	**0.5720**	**0.5731**	**0.5647**	**0.5706**	**0.5669**	**0.5634**	**0.5684**	**0.5678**
DCN V2	0.5771	0.5788	0.5663	0.5753	0.5651	0.5595	0.5669	0.5672
+DFFI	**0.5794**	**0.5816**	**0.5689**	**0.5774**	**0.5685**	**0.5630**	**0.5707**	**0.5692**
FiBiNET	0.5706	0.5718	0.5613	0.5691	0.5642	0.5576	0.5673	0.5627
+DFFI	**0.5769**	**0.5796**	**0.5630**	**0.5747**	**0.5690**	**0.5634**	**0.5714**	**0.5701**

5.4.2 *The compatibility of DFFI.*

In this part, we investigate the compatibility of DFFI with various feature interaction models. We apply it to four advanced feature interaction models, namely PNN [18], AutoInt [21], DCN V2 [25], FiBiNET [13]) on the Ali-CCP dataset and Ali-Mama dataset. As shown in Table 5, employing DFFI module to introduce domain knowledge to assist the feature interaction modeling achieves much better performance on these four models over both Ali-CCP and Ali-Mama datasets. Therefore, we can conclude that the DFFI module has a strong compatibility with feature interaction models.

6 CONCLUSION

In this study, we present the DFFM framework for multi-domain recommendations, which contains DFFI and DFUI modules. DFFI projects domain related features as deep neural networks placed on top of features. DFUI regards domain information as an auxiliary matrix of self-attention network to govern the modeling of user behaviors. Both offline and online experiments demonstrate the superiority of DFFM on multi-domain CTR prediction. Up to now, DFFM is deployed with major traffic in the Huawei advertising system, obtaining 4.13% improvement of ECPM.

ACKNOWLEDGMENTS

We thank the new deep learning computing framework MindSpore [1] for the partial support of this work.

REFERENCES

[1] 2020. MindSpore. *https://www.mindspore.cn.*

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79 (2010), 151–175.

[3] Steffen Bickel, Michael Brückner, and Tobias Scheffer. 2007. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning.* 81–88.

[4] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[5] Rich Caruana. 1997. Multitask learning. *Machine learning* 28 (1997), 41–75.

[6] Bo Chen, Yichao Wang, Zhirong Liu, Ruiming Tang, Wei Guo, Hongkun Zheng, Weiwei Yao, Muyu Zhang, and Xiuqiang He. 2021. Enhancing Explicit and Implicit Feature Interactions via Information Sharing for Parallel Deep CTR Models. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.* 3757–3766.

[7] Kun Gai, Xiaoqiang Zhu, Han Li, Kai Liu, and Zhe Wang. 2017. Learning piecewise linear models from large scale data for ad click prediction. *arXiv preprint arXiv:1704.05194* (2017).

[8] 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).

[9] Wei Guo, Rong Su, Renhao Tan, Huifeng Guo, Yingxue Zhang, Zhirong Liu, Ruiming Tang, and Xiuqiang He. 2021. Dual Graph enhanced Embedding Neural Network for CTR Prediction. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021.* ACM, 496–504.

[10] Wei Guo, Ruiming Tang, Huifeng Guo, Jianhua Han, Wen Yang, and Yuzhou Zhang. 2019. Order-aware Embedding Neural Network for CTR Prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019.* ACM, 1121–1124.

[11] Wei Guo, Can Zhang, Zhicheng He, Jiarui Qin, Huifeng Guo, Bo Chen, Ruiming Tang, Xiuqiang He, and Rui Zhang. 2022. Miss: Multi-interest self-supervised learning framework for click-through rate prediction. In *2022 IEEE 38th international conference on data engineering (ICDE).* IEEE, 727–740.

[12] Zhicheng He, Weiwen Liu, Wei Guo, Jiarui Qin, Yingxue Zhang, Yaochen Hu, and Ruiming Tang. 2023. A Survey on User Behavior Modeling in Recommender Systems. *arXiv preprint arXiv:2302.11087* (2023).

[13] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems.* 169–177.

[14] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/ARXIV.1412.6980

[15] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. AutoFIS: Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020.* ACM, 2636–2645.

[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 *SIGKDD.* 1930–1939.

[17] 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 *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval.* 1137–1140.

[18] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM).* IEEE, 1149–1154.

[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 *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.* 4104–4113.

[20] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 1161–1170.

[21] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 1161–1170.

[22] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *RecSys.* 269–278.

[23] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17.* ACM, 12.

[24] 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.

[25] Ruoxi Wang, Rakesh Shivanna, Derek Zhiyuan Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems. In *Proceedings of WWW*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 1785–1797.

[26] Bencheng Yan, Pengjie Wang, Kai Zhang, Feng Li, Hongbo Deng, Jian Xu, and Bo Zheng. 2022. Apg: Adaptive parameter generation network for click-through rate prediction. *Advances in Neural Information Processing Systems* 35 (2022), 24740–24752.

[27] Xuanhua Yang, Xiaoyu Peng, Penghui Wei, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. AdaSparse: Learning Adaptively Sparse Structures for Multi-Domain Click-Through Rate Prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management.* 4635–4639.

[28] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021. Deep learning for click-through rate estimation. *arXiv preprint arXiv:2104.10584* (2021).

[29] Guorui Zhou, Weijie Bian, Kailun Wu, Lejian Ren, Qi Pi, Yujing Zhang, Can Xiao, Xiang-Rong Sheng, Na Mou, Xinchen Luo, et al. 2020. CAN: Revisiting Feature Co-Action for Click-Through Rate Prediction. *arXiv preprint arXiv:2011.05625* (2020).

[30] 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.

[31] 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.

[32] Jie Zhou, Xianshuai Cao, Wenhao Li, Lin Bo, Kun Zhang, Chuan Luo, and Qian Yu. 2023. HiNet: Novel Multi-Scenario & Multi-Task Learning with Hierarchical Information Extraction. *arXiv preprint arXiv:2303.06095* (2023).

[33] Chenxu Zhu, Bo Chen, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2023. AIM: Automatic Interaction Machine for Click-Through Rate Prediction. *IEEE Trans. Knowl. Data Eng.* 35, 4 (2023), 3389–3403.