One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction

Xiang-Rong Sheng, Liqin Zhao*
Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo
Siran Yang, Jingshan Lv, Chi Zhang, Xiaoqiang Zhu
Alibaba Group
Beijing, China

ABSTRACT

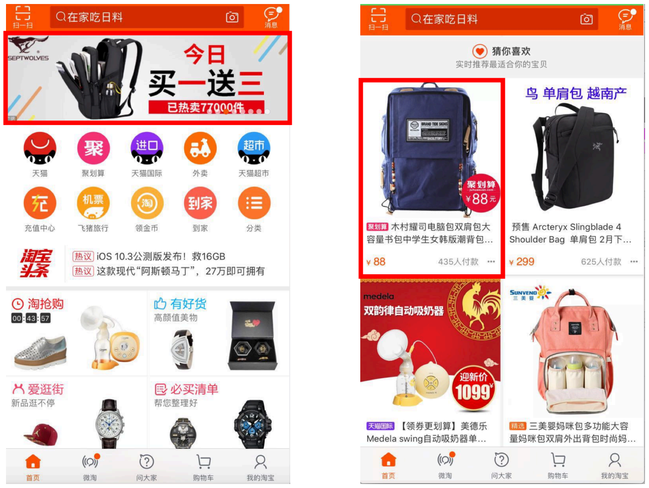
Traditional industrial recommenders are usually trained on a single business domain and then serve for this domain. In large commercial platforms, however, it is often the case that the recommenders need to make click-through rate (CTR) predictions for multiple business domains. Different domains have overlapping user groups and items, thus exist commonalities. Since the specific user group may be different and the user behaviors may change within a specific domain, different domains also have distinctions. The distinctions result in different domain-specific data distributions, which makes it hard for a single shared model to work well on all domains. To address the problem, we present Star Topology Adaptive Recommender (STAR), where one model is learned to serve all domains effectively. Concretely, STAR has the star topology, which consists of the shared centered parameters and domain-specific parameters. The shared parameters are used to learn commonalities of all domains and the domain-specific parameters capture domain distinction for more refined prediction. Given requests from different domains, STAR can adapt its parameters conditioned on the domain. The experimental result from production data validates the superiority of the proposed STAR model. Up to now, STAR has been deployed in the display advertising system of Alibaba, obtaining averaging 8.0% improvement on CTR and 6.0% on RPM (Revenue Per Mille).

CCS CONCEPTS

• **Information systems** → **Computational advertising**.

KEYWORDS

Click-Through Rate Prediction, Neural Networks, Display Advertising



Banner

Guess What You Like

Figure 1: Two representative business domains, *Banner* and *Guess What You Like* on Taobao mobile app home. A business domain is referred to as a specific spot that items are presented to users in the mobile app and PC websites.

ACM Reference Format:
Xiang-Rong Sheng, Liqin Zhao and Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Xiaoqiang Zhu. 2018. One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Traditional CTR prediction models [6, 12, 31, 39, 40] focuses on single-domain prediction, where the CTR model serves for a single business domain after trained with examples collected from this domain. Each business domain is a specific spot that items are presented to users on the mobile app or PC websites. At large commercial companies like Alibaba and Amazon, there are often a large number of business domains that need CTR prediction for enhancing user satisfaction. For example, in Alibaba, the business domains range from *Guess What You Like* in Taobao App homepage, *Banner* of Taobao App homepage to other little data domains like *Tian Mao Nong Chang* [42]. Figure 1 shows two representative business domains in Alibaba,

- *Banner*: In *banner*, the items to be recommended appears in the top banner of the Taobao home page. The item can be a single commodity, a store, or a brand.

*Xiang-Rong Sheng and Liqin Zhao contributed equally to this work. Corresponding author is Guorui Zhou. {xiangrong.sxr, liqin.zlq, guorui.xgr}@alibaba-inc.com

Woodstock '18, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

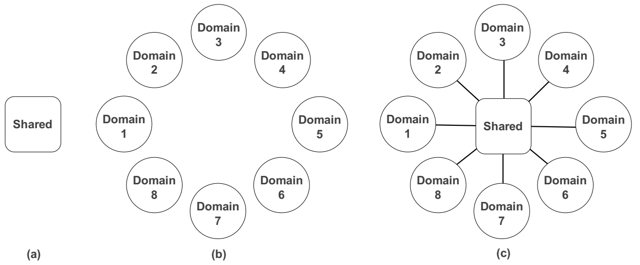


Figure 2: (a): Single shared model for all domains, square nodes indicate the shared model. (b): One model per domain where each model is learned separately. The circle node indicates the domain-specific model. (c): The proposed Star Topology Adaptive Recommender (STAR) where each domain has specific parameters and also shares a common centered model. The edges mean the combination of the center shared parameters with the domain-specific parameters.

- *Guess What You Like*: In *Guess What You Like*, items are all single commodities and displayed to users in the left or right column.

Since different business domains have overlapping user groups and items, there exist commonalities among these domains. Enabling information sharing among different domains is beneficial for learning the CTR model of each domain. For each business domain, however, the specific user group may be different and the users' behaviors may change in different domains. These distinctions result in different domain-specific data distributions. Simply mixing all the data and training a single shared CTR model may not work well on all domains.

Besides mixing data and training a shared model, another simple solution is to build a separate model per business domain. This strategy also has some downsides: (1) some business domains have much less data than other domains. Splitting the data neglects the domain commonalities and causes much less training data, which makes the complicated models hard to learn. (2) Maintaining multiple models simultaneously cause a huge amount of resource consumption and requires much more human cost. This is unduly burdensome when the number of business domains is up to hundreds.

In this paper, we aim to learn an effective and efficient CTR model to handle multiple domains simultaneously. We formulate *multi-domain CTR prediction* as the problem that the recommender needs to make CTR prediction for M business domains $D_1, D_2, \ldots, D_M$ simultaneously. The model takes input as $(\mathbf{x}, y, p)$, where $\mathbf{x}$ is the common feature used by multiple business domains like user historical behavior, user profile feature, item feature, and context feature. $y \in \{0, 1\}$ is the clicked label and p is the domain indicator that indicates which domain this sample is collected. Note that $(\mathbf{x}, y)$ is drawn from the domain-specific distribution D_p and the distribution varies with different domains. The goal of multi-domain CTR prediction is to construct an effective and efficient model that gives accurate CTR prediction for each domain and at a trivial cost on resource consumption. To achieve this goal, the models should make full use of the domain commonalities while captures the domain distinction.

One possible strategy to improve learning with multiple domains is multi-task learning [5, 32], where each domain can correspond to a task. The difference between multi-domain CTR prediction and multi-task learning is that multi-domain CTR prediction solves the same problem, i.e., CTR prediction, across different domains, in which the label space of different domains is the same. By contrast, multi-task learning addresses different problems in the same domain, where the label space might be different, e.g., jointly estimate CTR and CVR [24, 36]. Due to the heterogeneity of tasks, existing multi-task learning approaches focus on sharing information in the bottom layers but keeping separate task-specific output layers [32]. Directly adapting multi-task approaches to multi-domain CTR prediction cannot sufficiently exploit the domain relationship in the label space to enhance learning.

As mentioned above, existing approaches fail to utilize the homogeneity of label space of different domains to sufficiently model the domain commonalities and distinction. To fully exploit the domain relationship, we propose Star Topology Adaptive Recommender (**STAR**) to model multiple domains simultaneously. The proposed STAR model has the star topology, as illustrated in Figure 3. STAR consists of shared centered parameters and multiple sets of domain-specific parameters. The final model of each domain is obtained by combining the shared centered parameters and domain-specific parameters. The centered parameters are used to learn general behaviors among all domains, in which the common knowledge can be learned and transferred among all domains. The domain-specific parameters capture specific behaviors in different domains to facilitate more refined CTR prediction. The star topology facilitates effective information transformation across multiple domains to learn domain commonalities and distinction. In this paper, we implement the STAR model with the element-wise product of weights in each layer as the combination strategy. Since in industrial recommender, most parameters are contributed by embedding layers, the added domain-specific parameters are negligible to the total amount of parameters. Thus, using STAR model to serve multiple domains only adds little computational and memory costs while yielding much better performance.

The main contributions of this work can be summarized as follows:

- We propose Star Topology Adaptive Recommender (STAR) to tackle multi-domain CTR prediction. The star topology facilitates effective information transformation across multiple domains to learn domain commonalities while capturing domain distinction
- Data in different domains have different characteristics, which leads to inaccurate statistics when using batch normalization. To address this issue, we propose Partitioned Normalization (PN) that privatizes normalization for examples from different domains. PN can lead to more accurate moments within the domain, which improves model performance.
- In multi-domain CTR prediction, features that depict the domain information are of importance. We propose an auxiliary network that treats the domain indicator directly as the ID feature input and learns its embeddings to depict the domain. The embeddings are then fed to the auxiliary network, which is much simpler than the original network. This

makes the domain indicator influence the final prediction in a direct manner.

- We evaluate STAR on the industrial production dataset and deploy it in the display advertising system of Alibaba. The consistent superiority validates the efficacy of STAR. Up to now, the deployment of STAR brings 6% CTR and 8% RPM lift. We believe the lesson learned in our deployment generalize to other setups and are thus of interest to both researchers and industrial practitioners.

2 RELATED WORK

Our work is closely related to traditional single-domain CTR prediction, where the recommender is trained on a single business domain and then serve for this business domain. Besides, our work is also related to multi-task learning and multi-domain learning. In this section, we give a brief introduction.

2.1 Single-Domain CTR Prediction

Inspired by the success within deep learning, recent CTR prediction model has made the transition from traditional shallow approaches [11, 18, 19, 31, 41] to modern deep approaches [6, 12, 26, 29, 39, 40]. Most deep CTR models follow the embedding and MLP paradigm. Wide & Deep [6] and deepFM [12] combine low-order and high-order features to improve the expression power of the model. PNN [29] introduces a product layer to capture interactive patterns between inter-field categories. In these models, the user's history behaviors are transformed into low-dimensional vectors after the embedding and pooling. DIN [40] employs the mechanism of attention to activate historical behaviors locally w.r.t. the given the target item, and successfully captures the diversity characteristic of user interest. DIEN [39] further proposes an auxiliary loss to capture latent interest from historical behaviors. Additionally, DIEN integrates the attention mechanism with GRU to model the dynamic evolution of user interest. MIND [20] and DMIN [37] argues that a single vector might be insufficient to capture complicated pattern lying in the user and items. Capsule network and the dynamic routing mechanism are introduced in MIND to learn multiple representations to aggregate raw features. Moreover, inspired by the success of the self-attention architecture in the tasks of sequence to sequence learning [35], Transformer is introduced in [10] for feature aggregation. MIMN [27] proposes a memory-based architecture to aggregate features and tackle the challenge of long-term user interest modeling. SIM [28] extracts user interests with two cascaded search units, which achieves better ability to model lifelong sequential behavior data in both scalability and accuracy.

2.2 Multi-Task Learning

Multi-task learning [5, 32] aims to improve generalization by sharing knowledge across multiple related tasks. The shared knowledge and task-specific knowledge are explored to facilitate the learning of each task. Multi-task learning has been used successfully on multiple application domains, ranging from natural language processing [7] and speech recognition [8] to computer vision [16]. In early literature on MTL for linear models, Argyriou et al. [1] propose a method to learn sparse representations shared across multiple tasks. In the context of deep learning, multi-task learning is typically done with parameter sharing of hidden layers [5, 24]. Misra et al. [25] propose cross-stitch units to learn unique combinations of task-specific hidden-layers for each task. Ma et al. [23] proposes Multi-gate Mixture-of-Experts (MMoE) to model task relationships by sharing the expert sub-models across all tasks, while also having a gating network trained to optimize each task. Kendall et al. [16] propose a principled approach to multi-task deep learning which weighs multiple loss functions by considering the homoscedastic uncertainty of each task. In multi-task learning, different tasks may conflict, necessitating a trade-off, optimize a proxy objective that minimizes a weighted linear combination of per-task losses may not be optimal. To address this issue, Sener and Koltun [34] explicitly cast multi-task learning as multi-objective optimization, with the overall objective of finding a Pareto optimal solution. Note that [16, 34] are complementary to this work and could be potentially combined to achieve better performance.

2.3 Multi-Domain Learning

In real-world applications, it is oftentimes that the data are collected from multiple domains [9, 15, 21]. Multi-domain learning enables knowledge transfer between domains to improve learning. As such, it contrasts with the domain adaptation (DA) problem [3, 4], where knowledge transfer is only one way, i.e., from the source domain to the target domain. Multi-domain CTR prediction can be seen as a special kind of multi-domain learning problem, in which each domain corresponds to a business domain and the task is the CTR prediction. Compared with traditional multi-domain learning, our work focuses on CTR prediction. The proposed model makes full use of the domain indicator that is directly fed as the ID feature and learning its semantic embeddings to facilitates the model learning, which is neglected by previous literature. The difference between multi-domain learning and multi-task learning is that multi-domain learning makes prediction for multiple domains addressing the same problem, e.g., CTR prediction, where the label spaces are of the same. In contrast, multi-task learning focuses on tackling different problems [38]. For example, in the field of video recommendation, a multi-task learning problem can be as simultaneously predicting CTR and expected watch time of videos for a single business domain and multi-domain CTR prediction makes CTR predictions for multiple business domains, e.g., multiple video platforms.

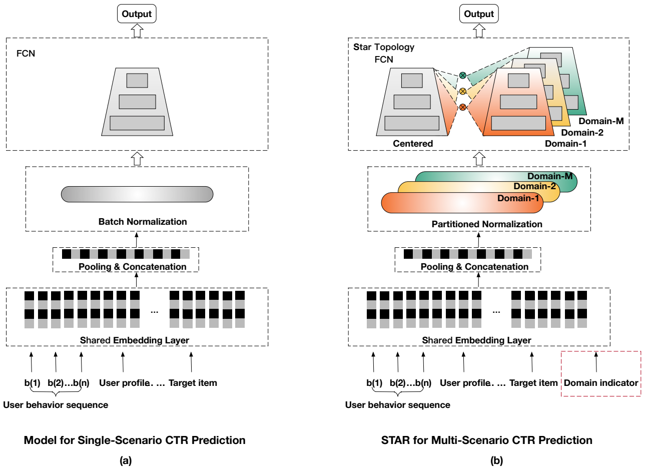
3 THE PROPOSED APPROACH

In this section, we first give a brief introduction about the background of multi-domain CTR prediction. Next is the architecture overview of the proposed method, star topology adaptive recommender (STAR) for multi-domain CTR prediction. Then we introduce STAR in detail, including the proposed star topology network, partitioned normalization, and auxiliary network.

3.1 Multi-Domain CTR Prediction

In sequential recommender systems, the model takes input as the user historical behavior, user profile feature, target item feature, and other features like context feature. The predicted CTR $\hat{y}$ of an user u clicking on an item m is calculated via:

$$\hat{y} = \mathrm{f}(E(u_1), \dots, E(u_i); E(m_1), \dots, E(m_j); E(c_1), \dots, E(c_k)),$$



Model for Single-Scenario CTR Prediction

(a)

STAR for Multi-Scenario CTR Prediction

(b)

Figure 3: Comparison of model for single-domain CTR prediction and the Star Topology Adaptive Recommender (STAR) for multi-domain CTR prediction. In STAR, the partitioned normalization (PN) privatizes normalization for examples from different domains. The normalized features are then fed as input to the following star topology fully-connected neural network (star topology FCN). The star topology FCN consists of shared centered FCN and multiple domain-specific FCNs. The final combined model of each domain is obtained by the element-wise product of weights in each layer.

where $\{u_1, \dots, u_i\}$ is the set of user features including user historical behavior and user profile feature. $\{m_1, \dots, m_i\}$ is the set of target item feature and $\{c_1, \dots, c_i\}$ is the set of other features. The $E(\cdot) \in \mathbb{R}^d$ means the embedding layer which maps the sparse IDs into learnable dense vector.

After mapping the raw features to low-dimensional embeddings, the common practice is to aggregate these embeddings to obtain fixed-length vectors. Different kinds of aggregation methods like [39, 40] can be employed to aggregate these embeddings to extract user interest and get the fixed-length representation. The obtained representation is then fed into the following deep neural network, e.g., a multi-layer fully-connected network, to get the final CTR prediction.

Traditional CTR models [6, 12, 22, 39, 40] are usually trained on data from a single business domain. However, real-world recommender often has to deal with multiple business domains. To enhance user satisfaction and promote advertising revenue, it is necessary to give accurate predictions for requests from all domains. Concretely, the recommender needs to make CTR prediction for M domains $D_1, D_2, \dots, D_M$ simultaneously. The model takes input as $(\mathbf{x}, y, p)$, where $\mathbf{x}$ is the common feature used by multiple business domains like user historical behavior and user profile feature, target item feature as mentioned above. $y \in \{0, 1\}$ is the clicked label and $p \in \{1, 2, \dots, M\}$ is the domain indicator that indicates which domain this sample is collected. Note that $(\mathbf{x}, y)$ is drawn from the domain-specific distribution D_p and the distribution varies for different domains. The goal of multi-domain CTR prediction is to construct a single CTR model that can give accurate CTR prediction to serve all domains at low resource consumption and human cost.

3.2 Architecture Overview

As mentioned above, ignoring domain indicator p and learning a single shared CTR model neglect the domain differences. This leads to inferior model performance. On the other hand, training separate models for each domain performs much worse since splitting the domains provides much less data for each model and makes it hard to fit complicated deep neural networks. Besides, due to the resource consumption and human cost, it is infeasible to maintain each domain a separate model in production.

To this end, we propose Star Topology Adaptive Recommender (STAR) for multi-domain CTR prediction to better utilize the similarity among different domains while capturing the domain distinction. As shown in Figure 3, STAR consists of three main components: (1) the **partitioned normalization** (PN) which privatizes normalization for examples from different domains, (2) the **star topology fully-connected neural network** (star topology FCN) which consists of shared centered FCN and multiple domain-specific FCNs. The final model of each domain is obtained by combining the shared FCN and domain-specific FCN. (3) the **auxiliary network** that treats the domain indicator directly as the input feature and learns its semantic embeddings to capture the domain distinction.

During training, a domain indicator p is first sampled and then a mini-batch of B instances

$$(\mathbf{x}_1, p), (\mathbf{x}_2, p), \dots, (\mathbf{x}_B, p)$$

is sampled from this domain. STAR first embeds these input features as low-dimensional vectors by an embedding layer. In industrial recommender, the model is often trained with billions of features [14] and the parameters of embedding are usually much more than other parts of the model. This makes it difficult for different domains to learn domain-specific embeddings with limited data. For example, for models used in our daily tasks, the embeddings parameters are 10,000 times more than the parameters of fully-connected layers [14]. Thus, in the proposed STAR model, we let all business domains share the same embedding layer, i.e., the same ID features in different domains share the same embedding. Sharing embedding layer across multiple domains can significantly reduce the computational and memory cost.

The embeddings are then pooled and concatenated to obtain B fixed-length representations. After that, the B extracted representations are processed by the proposed partitioned normalization (PN) layer that privatizes normalization statistics for different domains. During training, PN employs the batch statistics just as the batch normalization [13], but employs specific moments and parameters for different domains during testing. The normalized vectors are then fed as input to the proposed star topology FCN to get the output. The star topology FCN consist of shared centered FCN and multiple domain-specific FCNs. The final model of each domain is obtained by combining the shared centered FCN and domain-specific FCN, which is implemented by the element-wise product between weights in each layer. The purpose of the centered parameters is to learn general behaviors among all domains, in which the common knowledge is learned and transferred among all domains. The domain-specific parameters capture specific behaviors in different domains to facilitate more refined CTR prediction. The star topology enable effective information transformation across multiple domains to learn domain commonalities and distinction.

In multi-domain CTR prediction, features that depict the domain information is of importance. In STAR model, the auxiliary network treats the domain indicator as the ID feature. The domain indicator is first mapped into embedding and then fed with other features depicting the domain to the auxiliary network. The output of the auxiliary network is added with the output of the star topology FCN to get the final prediction. We make the auxiliary network much simpler than the star topology FCN to let the model capture the domain distinction in a direct and easy manner.

In what follows we will describe these components in detail.

3.3 Partitioned Normalization

As mentioned above, the raw features are first transformed into low-dimensional embeddings and then pooled and aggregated to get the intermediate representation. Denote the intermediate representation of an instance as $\mathbf{z}$, to train deep networks fast and stably, a standard practice is applying normalization layer to the intermediate representation $\mathbf{z}$. Among all normalization methods, batch normalization (BN) [13] is a representative method that is proved to be crucial to the successful training of very deep neural networks [13, 30]. BN use a global normalization for all examples, which accumulates normalization moments and learns shared parameters across all samples. Concretely, the normalization of BN in training is given as

$$\mathbf{z}' = \gamma \frac{\mathbf{z} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \tag{1}$$

where $\mathbf{z}'$ is the output, γ, β are the learnable scale and bias parameters, μ, σ^2 are mean and variances of current mini-batch. During testing, moving averaged statistics of mean E and variance Var across all samples are used instead

$$\mathbf{z}' = \gamma \frac{\mathbf{z} - E}{\sqrt{Var + \epsilon}} + \beta. \tag{2}$$

In other words, BN assume all samples are i.i.d. and use the shared statistics across all training samples.

However, in multi-domain CTR prediction, samples are only assumed to be locally i.i.d. within a specific domain. Thus, data from different domains have different normalization moments. Sharing global moments and parameters of BN layers during testing will obscure domain differences and lead to degraded model performance. To capture the unique data characteristic of each domain, we propose partitioned normalization (PN) which privatizes normalization statistics and parameters for different domains. Concretely, during training, suppose the current mini-batch is sampled from the p-th domain, we compute the mean and variances of the current mini-batch and normalize the feature as:

$$z' = (\gamma * \gamma_p) \frac{z - \mu}{\sqrt{\sigma^2 + \epsilon}} + (\beta + \beta_p), \tag{3}$$

where γ, β are the global scale and bias, and γ_p, β_p are the domain-specific scale and bias parameters. For each mini-batch, it receives the final scale by element-wise multiplying the shared γ with the domain-specific γ_p, i.e., PN adaptively scales the representation according to the domain indicator. Similarly, the bias of PN is also adaptive conditioned on the domain, which is implemented by the addition of global bias β and domain-specific bias β_p. Note that compared with BN, PN also uses the moments of the current mini-batch during training, but PN introduces domain-specific scale and bias γ_p, β_p to capture the domain distinction.

Besides the modification of the scale and bias, PN also let different domains to accumulate the domain-specific moving average of mean E_p and variance Var_p. During testing, PN transforms instance $\mathbf{z}$ from the p-th domain as:

$$\mathbf{z}' = (\gamma * \gamma_p) \frac{\mathbf{z} - E_p}{\sqrt{Var_p + \epsilon}} + (\beta + \beta_p). \tag{4}$$

From Equation 4, we can see that PN use the domain-specific mean E_p and variance Var_p to normalize the intermediate representation $\mathbf{z}$. Thus PN adaptively alters the intermediate representation conditioned on the domain indicator to capture the distinctive characteristics.

3.4 Star Topology FCN

After the PN layer, the representation $\mathbf{z}'$ is fed as input to the following star topology multi-layer fully-connected neural network (star topology FCN). As depicted in Figure 3, the proposed star topology FCN consists of a shared centered FCN and independent FCNs per domain, thus the total number of FCN is $M + 1$. The final model of p-th domain is obtained by combining the shared centered FCN and domain-specific FCN, in which the centered parameters learn general behaviors among all domains, and the domain-specific parameters capture specific behaviors in different domains to facilitate more refined CTR prediction.

Specifically, for the shared FCN, let W be the weights and b be the bias in a neural network layer respectively. For the specific FCN of the p-th domain, let W_p be the weights and b_p be the bias in the corresponding layer. Denote the input dimension as c and the output dimension as d, i.e, $W, W_p \in \mathbb{R}^{c \times d}, b, b_p \in \mathbb{R}^d$. The final weights $W_i^\star$ and bias $b_i^\star$ for the p-th domain is obtained by:

$$W_p^\star = W_p \otimes W, b_p^\star = b_p + b, \tag{5}$$

where $\otimes$ denotes the element-wise multiplication. Let $in_p \in \mathbb{R}^{c \times 1}$ denote the input of this neural network layer from the p-th domain, the final output $out_p \in \mathbb{R}^d \times 1$ is given by:

$$out_p = \phi((W_p^\star)^\top in_p + b_p^\star), \tag{6}$$

where ϕ denotes the activation function of this layer. The combination of shared parameters of domain-specific parameters is employed in all layers. By this means, STAR can modulate its parameters conditioned on the domain.

Note that we implement the combination strategy of the shared centered FCN and domain-specific FCN by element-wise product of between weights and addition of bias in each layer, other strategies can also be investigated for better performance. Note that the shared parameters are updated through the gradient of all examples while the domain-specific parameters are only updated through examples within this domain. This helps captures the domain differences for more refined CTR prediction while learning the domain commonality through the shared centered parameters. As mentioned above, most of the parameters in industrial recommenders are contributed by the embedding layer, the increased M FCNs is negligible to the total amount of parameters. Thus STAR uses one model to effectively serve all business domains in a parameter efficient and memory friendly manner.

3.5 Auxiliary Network

In the traditional way of CTR modeling, all features are treated equally and fed to the complicated model. In multi-domain CTR prediction, however, it may be hard for the model to automatically learning the domain distinction. We argue that a good multi-domain CTR model should have the following characteristic: (1) have informative features regarding the domain characteristic (2) make these features easily and directly influence the final CTR prediction. The intuition behind is that features that depict the information of domains are of importance since it can reduce the difficulty for the model to capture the distinction among domains.

To this end, we propose an auxiliary network to learn the domain distinction. To augment informative features regarding the domain characteristic, we treat the domain indicator directly as the ID feature input. The domain indicator is first mapped into embedding vector and concatenated with other features. The auxiliary network then computes forward pass with respect to the concatenated features to gets its output. Denote the output of star topology FCN as s_m and the output of the auxiliary network as s_a. s_m and s_a are added to get the final logits. Then we apply softmax to the final logits and get the CTR prediction as:

$$\text{softmax}(s_m + s_a). \tag{7}$$

In our implementation, the auxiliary network is much simpler than the main network, which is a two-layer fully connected neural network. The simple architecture makes the domain features directly influence the final prediction.

Denote $\hat{y}_i^p$ the predicted probability for the i-th instance in the p-th domain and $y_i^p \in \{0, 1\}$ the ground truth. We minimize the cross entropy loss function between the $\hat{y}_i^p$ and label y_i^p in all domains as:

$$\min \sum_{p=1}^{M} \sum_{i=1}^{N_p} -y_i^p \log(\hat{y}_i^p) - (1 - y_i^p) \log(1 - \hat{y}_i^p). \tag{8}$$

4 EXPERIMENTS

We evaluate the efficacy of STAR in this section. We begin by introducing the setup including the used production dataset, compared methods and implementation details in Sec. 4.1. The results and discussion are elaborated in Sec. 4.2. We also perform in-depth ablation studies in Sec. 4.3. Experimental results on production environment are shown in Sec. 4.4.

4.1 Experimental Settings

Dataset. Due to the lack of public dataset on multi-domain CTR prediction, we use Alibaba production data regarding user click behavior on 19 business domains to perform the offline evaluation. The training data is collected from traffic logs of the online display advertising system of Alibaba. Data of one day from 19 business domains are used for training and the data of the following day is used for testing. The training dataset consists of billions of examples. Table 1 shows the example percentage and average CTR (# Click/# Impression, i.e., ratio of positive examples) of each domain in the training set. As shown in Table1, different domains have different domain-specific data distribution, which can be reflected from the different CTR. It can be seen that domain with the highest CTR (domain #15) is 12.03% while domain with the lowest CTR (domain #13) is only 1.27%.

Compared models. To verify the effectiveness of the proposed approach, we compare STAR with the following models:

- **Base**. We refer to Base as the model composed of embedding layer, pooling & concatenation layer, batch normalization, and a 7-layer fully-connected network. Specifically, the pooling & concatenation layer is based on DIEN [39], which extracts user interest after the embedding layer. We mix all samples from different domains and train the base model.
- **Shared Bottom**. The Shared Bottom model is a multi-task model that shares the parameters of the bottom layers. In our implementation, we let the Shared Bottom share the embedding layer. Each domain will also have a specific 7-layer fully-connected network that is not shared.
- **MulANN**. MulANN [33] adds domain discriminator module to the Base model. The domain discriminator classifies which domain the examples are from. MulANN adopts a adversarial loss to let the domain discriminator indiscriminates with respect to the shift between the domains.
- **MMoE**. MMoE [23] implicitly models task relationships for multi-task learning, where different tasks may have different label spaces. Here we adapt MMoE for multi-domain CTR prediction, where each fully-connected network is treated as an expert. The number of experts is equal to the number of domains. Besides, MMoE also learns gating networks per domain that takes the input features and outputs softmax gates assembling the experts with different weights.
- **Cross-Stitch**. Cross-Stitch [25] uses linear cross-stitch units to learn an optimal combination of task-specific representations. In the cross-stitch method, each domain have a 7-layer fully-connected network and the cross-stitch units are added in each hidden layer to learn task-specific representations.

To give a fair comparison, all compared methods and the STAR model are trained with the proposed auxiliary network in Sec. 4.2. The ablation study about the auxiliary network is performed in Sec. 4.3.

Implementation details. All models are trained with Adam [17], the learning rate is set to 0.001 and the batch size is 2000. We minimize the cross-entropy loss for samples from all domains to train the model.

Metrics. Area under the ROC curve (AUC) is the common metric used to evaluate the performance of CTR prediction. An variation of user weighted AUC [40] measures the goodness of intra-user order by averaging AUC over users and is shown to be more relevant to online performance in recommender system. It is calculated as follows:

$$\text{AUC} = \frac{\sum_i^n \#\text{impression}_i \times \text{AUC}_i}{\sum_i^n \#\text{impression}_i}, \tag{9}$$

where n is the number of users, impression$_i$ and AUC$_i$ are the number of impressions and AUC of the i-th user, respectively. We use this weighted AUC as the evaluation metric and still refer it to

Table 1: The example percentage and average click-through rate (CTR) of each domain.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Percentage	0.99%	1.61%	3.40%	3.85%	2.79%	0.56%	4.27%	16.76%	10.00%	12.16%	0.76%	1.31%	3.34%	28.76%	1.17%	0.46%	1.05%	0.91%	5.85%
CTR	2.14%	2.69%	2.97%	3.63%	2.77%	3.45%	3.59%	3.24%	3.23%	2.08%	12.05%	3.52%	1.27%	3.75%	12.03%	4.02%	1.63%	4.64%	1.42%

as AUC for simplicity. Concretely, we use the AUC of each domain and overall AUC (mixing samples from all domains to calculate the overall AUC) as the metrics.

4.2 Results

We evaluate all approaches on the Alibaba production dataset. To give a fair comparison, all compared methods and STAR model are trained with the proposed auxiliary network. As illustrated in Table 2, the consistent improvement validates the efficacy of STAR. Note that the performance of MulANN is worse than the Base model, which proves obscuring domain difference hurts the modeling of multi-domain CTR prediction. Besides, the shared Bottom model, MMoE, Cross-Stitch and STAR all achieve better overall performance than the Base model. This demonstrates the importance of exploiting domain relationship and capturing domain distinction to enhance the prediction performance.

Although the Shared Bottom, MMoE, and Cross-Stitch achieve better overall performance than the Base model, it is notable that in some domains, the AUCs of Shared Bottom, MMoE, and Cross-Stitch are worse than the Base model, e.g., domain # 5, #6, and #16. In contrast, the proposed STAR model exhibits superior performance across all domains compared with the Base model. Besides, STAR also achieves consistent improvement than the Shared Bottom, which demonstrates the importance of information sharing on top specific layers for multi-domain learning, where all domains share the same label space. STAR also outperforms MMoE and Cross-Stitch, which shows the superiority of explicitly model domain relationships compared with implicitly modeling domain relationships by the gate networks or cross-stitch units.

Table 2: Results of different approaches on offline Alibaba production dataset.

	Base	Shared Bottom	MulANN	MMoE	Cross-Stitch	STAR
#1	0.6134	0.6186	0.6143	0.6143	0.6183	**0.6306**
#2	0.6321	0.6320	0.6321	0.6355	0.6337	**0.6417**
#3	0.6281	0.6293	0.6282	0.6311	0.6307	**0.6372**
#4	0.6326	0.6361	0.6333	0.6373	0.6372	**0.6451**
#5	0.6308	0.6292	0.6302	0.6336	0.6322	**0.6388**
#6	0.6378	0.6383	0.6336	0.6412	0.6368	**0.6494**
#7	0.6305	0.6329	0.6310	0.6340	0.6352	**0.6410**
#8	0.6297	0.6278	0.6297	0.6330	0.6328	**0.6411**
#9	0.6264	0.6283	0.6258	0.6292	0.6278	**0.6368**
#10	0.6392	0.6434	0.6375	0.6431	0.6278	**0.6577**
#11	0.6469	0.6529	0.6445	0.6508	0.6548	**0.6719**
#12	0.6506	0.6575	0.6498	0.6518	0.6570	**0.6676**
#13	0.6558	0.6612	0.6538	0.6603	0.6637	**0.6739**
#14	0.6362	0.6405	0.6371	0.6412	0.6411	**0.6486**
#15	0.6745	0.6888	0.6710	0.6787	0.6819	**0.7021**
#16	0.6638	0.6627	0.6517	0.6634	0.6727	**0.6901**
#17	0.6524	0.6658	0.6499	0.6519	0.6575	**0.6715**
#18	0.6493	0.6480	0.6375	0.6500	0.6610	**0.6754**
#19	0.6330	0.6375	0.6306	0.6374	0.6381	**0.6476**
Overall AUC	0.6364	0.6398	0.6353	0.6403	0.6415	**0.6506**

4.3 Ablation Study

To investigate the effect of each component, we conduct several ablation studies.

4.3.1 STAR Topology FCN and PN. We analyze the influence of different components of STAR. Concretely, the separate effects of star topology FCN and PN are investigated. We compare (a) the Base model trained with BN, (b) Base model trained with PN, (c) STAR FCN with BN and (d) STAR model (STAR FCN + PN). The result is reported in Table 3. We observe that using star topology FCN and PN separately can outperform the Base model. Bring them together can further boost performance. The result validates the effect of both star topology FCN and PN.

Table 3: Ablation study of partitioned normalization (PN) and star topology fully-connected neural networks (STAR FCN). All models are trained with the proposed auxiliary network.

	Base (BN)	Base (PN)	STAR FCN (BN)	STAR FCN (PN)
Overall AUC	0.6364	0.6485	0.6415	**0.6506**

4.3.2 Normalization. Normalization methods are very effective components in deep learning, which have been shown by many practices to ease optimization and enable very deep networks to converge. We analyze the effect of different normalization methods including Batch Normalization (BN) [13], Layer Normalization (LN) [2] and the proposed Partitioned Normalization (PN) on multi-domain CTR prediction. BN accumulates global statistics and learns global parameters for samples from all domains. LN is a representative instance-based normalization method, which operates along the channel dimension and avoids mixing statistics for samples from different domains.

The result is shown in Table 4. Our first observation is that both LN and PN outperforms BN. This observation validates that data from different domains have distinct distribution and need specific normalization. Using global normalization obscures domain differences, which will hurt performance for multi-domain CTR prediction. We also observe that PN outperforms LN, which validates that domain-specific normalization is better than the instance-specific normalization, since PN leads to more accurate moments within the domain.

4.3.3 Auxiliary network. We conduct experiment to assess the effect of the auxiliary network for different models. All methods are trained with and without the proposed auxiliary network. The result is illustrated in Figure 4. We observe that the auxiliary network

Table 4: Ablation study of normalization methods for multi-domain CTR prediction. STAR FCN is trained BN, LN, and PN respectively.

	STAR FCN (BN)	STAR FCN (LN)	STAR FCN (PN)
Overall AUC	0.6415	0.6463	**0.6506**

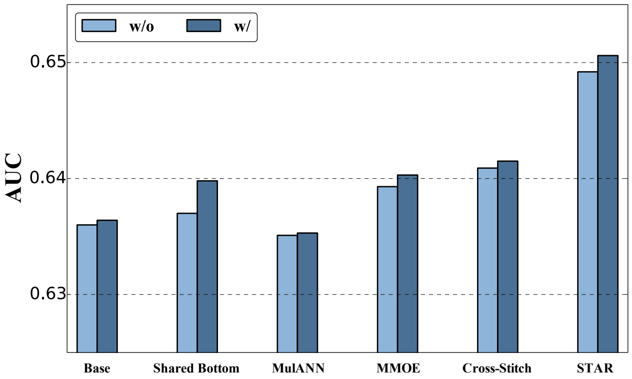


Figure 4: The performance of different methods trained with (w/) and without (w/o) the auxiliary network.

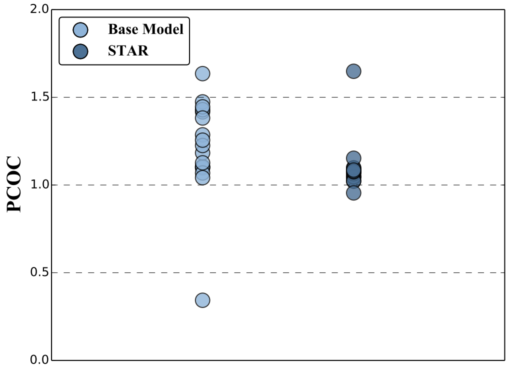


Figure 5: Predicted CTR over CTR (PCOC) of the Base model and STAR in all domains. Each circle means PCOC of a specific domain.

Table 5: CTR and RPM gains in online display advertising system of Alibaba.

	CTR	RPM
Overall	+8.0%	+6.0%

improves all methods consistently. The result validates the importance of making full utilization of domain features and using it to capture the domain distinction. We also observe the improvement of the auxiliary network for MulANN is slightly weaker than the other methods. The reason may due to the fact that the adversarial loss for obscuring domain differences contradicts with the domain feature to capture the domain differences.

4.3.4 Ability to Capture Domain Distinction. We show that STAR is more capable of capturing domain distinction. We first compute the predicted CTR over CTR (PCOC) of the Base model and STAR in each domain. Note that the closer PCOC is to 1.0, the more accurate the CTR prediction is. The PCOCs is shown in Figure 5. We can see that the PCOCs of STAR in different domains are more compact and concentrated around 1.0 compared with the Base model. This validates the ability of STAR to capture the domain distinction.

4.4 Production

Online serving and challenges. One of the challenges in industrial recommender is that the distribution of features and CTR exhibits large shifts over time. To capture the dynamic change of data in real-time, it is important to use real-time examples to update the CTR models continuously to prevent them from becoming stale. However, for multi-domain CTR prediction, the percentage of examples of each domain changes over time. For example, some business domains have traffic spike in the morning while some business domains have traffic spike in the evening. If we train the model directly in the chronological order, the changes in data percentage over time will cause the instability of model learning. To address this issue, we redesign the data pipeline and maintain a buffer that stores a sliding window of history samples to avoid the sudden change of example percentage. Specifically, samples in the buffer are shuffled firstly and then sampled to construct a mini-batch. After fed to the model, this mini-batch of samples are removed from the buffer and new arriving data is added to this buffer. We empirically found this training manner is more stable than the traditional way of online updates.

The systematical optimization makes STAR capable of serving main traffic of multiple business domains stably. Table 5 shows the improvement of STAR on the display advertising system of Alibaba, where STAR is deployed and serves for more than 60 business domains. We compute the overall improvements of all domains. The introduction of STAR brings +8.0% overall CTR lift and +6.0% overall RPM lift in our online A / B test compared with the Base model.

5 CONCLUSION

In this paper, we propose the star topology adaptive recommender to address the problem of multi-domain CTR prediction. Instead of keeping unique models for different domains or simply mixing all samples and maintaining a shared model, STAR has the star topology, which consists of shared centered parameters and domain-specific parameters. The shared parameters learn commonalities, which is updated through all examples. The domain-specific parameters capture domain distinction for more refined prediction, which is learned using examples within the specific domain. By this means, STAR can adaptively modulate its parameters conditioned on the domain for more refined prediction. The experiments demonstrate that the superiority of STAR on multi-domain CTR prediction. Up to now, STAR is deployed in the advertising system of Alibaba, obtaining 8.0% improvement on CTR and 6.0% on RPM.

REFERENCES

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex multi-task feature learning. *Machine Learning* 73, 3 (2008), 243–272.

[2] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *CoRR* abs/1607.06450 (2016).

[3] 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, 1-2 (2010), 151–175.

[4] 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*, Vol. 227. 81–88.

[5] Rich Caruana. 1998. Multitask Learning. In *Learning to Learn*. 95–133.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.

[7] Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25-th International Conference on Machine Learning*, William W. Cohen, Andrew McCallum, and Sam T. Roweis (Eds.), Vol. 307. 160–167.

[8] Li Deng, Geoffrey E. Hinton, and Brian Kingsbury. 2013. New Types of Deep Neural Network Learning for Speech Recognition and Related Applications: An Overview. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 8599–8603.

[9] Mark Dredze, Alex Kulesza, and Koby Crammer. 2010. Multi-Domain Learning by Confidence-Weighted Parameter Combination. *Maching Learning* 79, 1-2 (2010), 123–149.

[10] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-Through Rate Prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2301–2307.

[11] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. Melbourne, Australia., 2782–2788.

[13] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37. 448–456.

[14] Biye Jiang, Chao Deng, Huimin Yi, Zelin Hu, Guorui Zhou, Yang Zheng, Sui Huang, Xinyang Guo, Dongyue Wang, Yue Song, et al. 2019. XDL: An Industrial Deep Learning Framework for High-Dimensional Sparse Data. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. 1–9.

[15] Mahesh Joshi, Mark Dredze, William W. Cohen, and Carolyn Penstein Rosé. 2012. Multi-Domain Learning: When Do Domains Matter?. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 1302–1312.

[16] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. 7482–7491.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

[18] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, Nevada, USA, 426–434.

[19] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.

[20] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.

[21] Pengcheng Li, Runze Li, Qing Da, Anxiang Zeng, and Lijun Zhang. 2020. Improving Multi-Scenario Learning to Rank in E-commerce by Exploiting Task Relationships in the Label Space. In *Proceedings of The 29th ACM International Conference on Information and Knowledge Management0*. 2605–2612.

[22] 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 *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London, United Kingdom.

[23] 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 *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1930–1939.

[24] 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.

[25] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-Stitch Networks for Multi-task Learning. 3994–4003.

[26] 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*. Anchorage, AK, USA, 2671–2679.

[27] 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*. 1059–1068.

[28] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction. In *Proceeding of The 29th ACM International Conference on Information and Knowledge Management*. 2685–2692.

[29] 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 *Proceedings of the16th International Conference on Data Mining*. IEEE, 1149–1154.

[30] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *Proceedings of the 4th International Conference on Learning Representations*.

[31] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the 10th International Conference on Data Mining*. IEEE, 995–1000.

[32] Sebastian Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *CoRR* abs/1706.05098 (2017).

[33] Alice Schoenauer Sebag, Louise Heinrich, Marc Schoenauer, Michèle Sebag, Lani F. Wu, and Steven J. Altschuler. 2019. Multi-Domain Adversarial Learning. In *Proceedings of the 7th International Conference on Learning Representations*.

[34] Ozan Sener and Vladlen Koltun. 2018. Multi-Task Learning as Multi-Objective Optimization. In *Advances in Neural Information Processing Systems 31*. 525–536.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*. 5998–6008.

[36] 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 *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 2377–2386.

[37] 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 and Knowledge Management*. 2265–2268.

[38] Yongxin Yang and Timothy M. Hospedales. 2015. A Unified Perspective on Multi-Domain and Multi-Task Learning. In *Proceeding of the 3rd International Conference on Learning Representations*.

[39] 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 33rd AAAI Conference on Artificial Intelligence*. Honolulu, Hawaii, USA, 5941–5948.

[40] 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*. ACM, 1059–1068.

[41] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. 2008. Large-scale parallel collaborative filtering for the netflix prize. In *Proceedings of the International Conference on Algorithmic Applications in Management*. Springer, 337–348.

[42] 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 *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2191–2200.