

# Continual Transfer Learning for Cross-Domain Click-Through Rate Prediction at Taobao

Lixin Liu<sup>1</sup>, Yanling Wang<sup>2</sup>, Tianming Wang<sup>1</sup>, Dong Guan<sup>1</sup>  
Jiawei Wu<sup>1</sup>, Jingxu Chen<sup>1</sup>, Rong Xiao<sup>1</sup>, Wenxiang Zhu<sup>1</sup>, Fei Fang<sup>1</sup>

<sup>1</sup> Alibaba Group, <sup>2</sup> School of Information, Renmin University of China

Beijing, China

{llx271805,tianming.wtm,jiexing.gd,wujiawei,xianlu,wenxiang.zwx,mingyi.ff}@alibaba-inc.com

wangyanling@ruc.edu.cn,xiaorong.xr@taobao.com

## ABSTRACT

As one of the largest e-commerce platforms in the world, Taobao's recommendation systems (RSs) serve the demands of shopping for hundreds of millions of customers. Click-Through Rate (CTR) prediction is a core component of the RS. One of the biggest characteristics of CTR prediction at Taobao is that there exist multiple recommendation domains where the scales of different domains vary significantly. Therefore, it is crucial to perform cross-domain CTR prediction to transfer knowledge from large domains to small domains to alleviate the data sparsity issue. However, existing cross-domain CTR prediction methods are proposed for static knowledge transfer, ignoring that all domains in real-world RSs are continually time-evolving. In light of this, we present a necessary but novel task named Continual Transfer Learning (CTL), which transfers knowledge from a time-evolving source domain to a time-evolving target domain. In this work, we propose a simple and effective CTL model called CTNet to solve the problem of continual cross-domain CTR prediction at Taobao, and CTNet can be trained efficiently. Particularly, CTNet considers an important characteristic in the industry that models have been continually well-trained for a very long time. So CTNet aims to fully utilize all the well-trained model parameters in both source domain and target domain to avoid losing historically acquired knowledge, and only needs incremental target domain data for training to guarantee efficiency. Extensive offline experiments and online A/B testing at Taobao demonstrate the efficiency and effectiveness of CTNet. CTNet is now deployed online in the recommender systems of Taobao, serving the main traffic of hundreds of millions of active users.

## CCS CONCEPTS

• Information systems → Information retrieval.

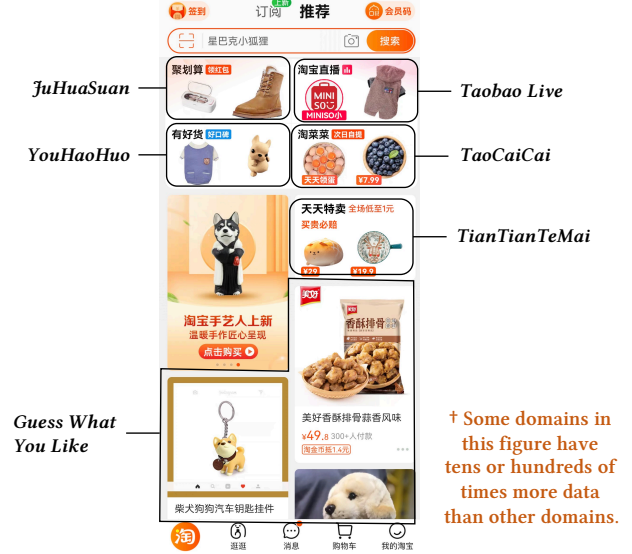
## KEYWORDS

Continual Transfer Learning, Recommender Systems, Cross-Domain Click-Through Rate Prediction, Click-Through Rate Prediction

## 1 INTRODUCTION

Click-through rate (CTR) prediction [5, 22, 28, 35, 36] which estimates the probability of a user to click on a candidate item, is a crucial task for real-world recommender systems (RSs), search engines, and online advertising systems. In Taobao<sup>1</sup>, one of the largest e-commerce platforms in the world, there are multiple recommendation domains where each domain has its own CTR prediction model.

<sup>1</sup><https://www.taobao.com>



**Figure 1: Entrances of different recommendation domains in Taobao mobile app home. Users can click the icon of a domain to browse their own recommendation feed lists.**

Figure 1 shows an example of multiple recommendation domains in Taobao mobile app. A user can easily browse across domains. Different domains have their specific themes (such as collections of good product, new items, specials, live streaming, etc.), and also share some items. In reality, the scales of user-item interactions in different domains can vary significantly, so it is useful to apply the techniques of cross-domain CTR prediction [12, 15, 16, 21], which transfer knowledge from the larger source domain to enhance CTR prediction on the smaller target domain.

Previous efforts on cross-domain CTR prediction can be broadly categorized into joint learning methods [16, 18, 21] and pre-training & fine-tuning methods [1]. The former ones jointly optimize the target domain model and the source domain model. Shared model parameters such as user embeddings and item embeddings establish the connection between different domains to enable knowledge transfer. However, such methods are often inefficient to deploy online at large portals like Taobao due to the very large-scale source domain training data. Besides, multi-task learning could induce negative impacts to the target domain task because of the conflicts between source domain objectives and target domain objectives [1, 30]. Alternatively, pre-training & fine-tuning methods replace target

**Table 1: Comparison between existing cross-domain CTR prediction methods and the proposed CTNet.**

Method	Free from the source domain data?	Not affected by source domain training objectives?	Only need incremental data for training?
Joint learning methods	No	No	Yes
Pre-training & fine-tuning methods	Yes	Yes	No
CTNet (Ours)	Yes	Yes	Yes

domain model parameters with pre-trained source domain model parameters and then fine-tune the target domain model. Since only target domain data and target domain objectives are used for optimization, pre-training & fine-tuning methods are usually more efficient and effective for industrial applications [1].

In fact, CTR prediction in real-world systems including Taobao follows a continual learning setting, where both the source domain and target domain are time-evolving. Each model gradually extends acquired knowledge by consistently training with new data, such that the latest user interests can be captured. However, existing methods are proposed for static knowledge transfer ignoring the continual setting. In light of this, we study cross-domain CTR prediction under the setting of continual learning, which is an important but less-explored task in both academia and industry. Specifically, we define a novel task named **Continual Transfer Learning (CTL)** which transfers knowledge from a time-evolving source domain to a time-evolving target domain. A straightforward solution of CTL is performing existing pre-training & fine-tuning methods every time the source domain model is updated by new data. Compared with offline training on small-scale benchmark datasets, a special characteristic of training for industrial systems is that **the current online models have been continually well-trained for years**. So we need to fully utilize the parameters of both the source domain model and the target domain model. Directly using the pre-training & fine-tuning methods, previously well-trained parameters of the target domain model are directly discarded and replaced by that of the source domain model, so a large amount of historical target domain data is required to guarantee the quality of every target domain fine-tuning, making CTL impractical. Furthermore, as the fine-tuning data from target domain is still relatively large (e.g. billions of samples), the knowledge from the source domain models could be forgotten during the fine-tuning. Therefore, it is necessary to design a more efficient CTL model for the applications at Taobao.

In this paper, we propose a simple and effective approach called **Continual Transfer Network (CTNet)** for CTL in the task of CTR prediction at Taobao. Distinct from existing pre-training & fine-tuning methods, CTNet preserves all the target domain model parameters which have been well-trained by the historical target domain data. Particularly, we design a light-weighted adapter module to map the intermediate layer hidden representations in the source domain model to be external knowledge for target domain training. In doing this, CTNet only needs incremental target domain data for training and can effectively adapt to the newest user-item interactions from both the source domain and target domain.

We provide the comparison between existing cross-domain CTR prediction methods and the proposed CTNet in Table 1, when deploying on large-scale platforms like Taobao. CTNet combines the advantages of joint learning methods and pre-training & fine-tuning methods, and avoids their disadvantages. First, CTNet is a source data-free model which performs efficiently when the size of source domain training data is very large. Second, only the target domain objective function is used for optimization, so CTNet is not affected by the the source domain training. Additionally, benefiting from the continually well-trained target domain model, CTNet just requires incremental target domain data for efficient CTL.

To sum up, the main contributions of this work include:

- We study an important but novel problem – continual transfer learning (CTL). Not limited to the CTR prediction task, the proposed CTL task can have broader applications in other real-world scenarios.
- We present CTNet, a simple and effective CTL model for continual cross-domain CTR prediction at Taobao. CTNet continually makes use of the well-trained source domain model and target domain model, and only requires the incremental target domain training data, such that it can perform efficiently for large-scale continual transfer learning.
- We conduct extensive offline experiments and online A/B testing at Taobao. In addition to its high efficiency, CTNet shows superior performance compared with the state-of-the-art methods.
- Since December 2021, CTNet has been deployed online in recommender systems of Taobao serving the main traffic, and consistently brings significant performance improvement. We provide our experience of successfully deploying deep transfer learning models in large-scale recommender systems, which will be helpful and practical for many industrial applications.

## 2 RELATED WORK

This work is closely related to click-through rate prediction, cross-domain recommendation, transfer learning, and continual learning.

### 2.1 Click-Through Rate Prediction

Click-through rate (CTR) prediction [33] plays a core role in industrial online services including recommender systems, web search, and online advertising. In the deep learning era, a variety of powerful models are proposed for the task. Methods like Wide & Deep [5], DeepFM [10], and DCN [28] capture the high-order feature interactions by deep neural networks. In order to model massive user behaviors, methods including DIN [36], DIEN [35], BST [4], SIM [22], and ETA [2], etc, utilize various attention-based models

to model user behavior sequences. However, these methods are proposed for single-domain CTR prediction.

## 2.2 Cross-Domain Recommendation

This paper focus on the cross-domain CTR prediction, which can be naturally solved by some techniques of a related field – cross-domain recommendation (CDR). CDR is proposed to transfer knowledge across domains for solving the cold-start and data sparsity issues. Recent CDR approaches can be broadly divided into joint learning [12, 15, 16, 18, 19, 21, 37] methods and pre-training & fine-tuning [1, 31] methods. In section 1, we have analyzed key characteristics of existing CDR methods. Note that not all CDR methods are CDCTR methods. Since the CTR prediction task is very different from some other recommendation tasks like sequential recommendation [20], user modeling [31, 32], etc., so that many CDR methods can not be applied to the cross-domain CTR prediction task.

## 2.3 Transfer Learning

CDR models have been inspired a lot by the advances in transfer learning. Transfer learning [39] improves the learning performance on the target domain using the knowledge from the source domain. Recent methods on parameter-efficient transfer learning [11, 31] fine-tune a small part of pre-trained parameters while keeping most pre-trained parameters frozen. Distinctly, our model keeps all the pre-trained source domain parameters frozen, only using the hidden representations of pre-trained models as auxiliary knowledge to improve target domain training. A concurrent study Head2Toe [7] directly inputs intermediate representations of pre-trained models to target domain linear predictor. Instead of directly using the intermediate representations, we design an light-weighted adapter to summarize the hidden representations as the auxiliary knowledge to serve the target domain model training.

Recent attempts such as continual pre-training [13, 24] in NLP and continual domain adaptation [8, 17, 26, 27, 29], perform transfer learning under the continual learning setting, which seems similar to our work. It is worth noting that, in the settings of continual pre-training and continual domain adaptation, only one of the source domain and the target domain is time-evolving. Conversely, both source domain and the target domain are continually time-evolving in setting of our proposed continual transfer learning.

## 2.4 Continual Learning

Continual learning (CL) [6, 14, 23], also referred to as lifelong learning or incremental learning, aims to learn from an infinite stream of data, with the goal of gradually extending acquired knowledge and using it for future learning [14]. Research in this field mainly focuses on solving the catastrophic forgetting problem, i.e., performances on old tasks significantly degrade with the training of new tasks. This work does not focus on the catastrophic forgetting problem, since we hope the recommendation models can better serve the future prediction. Therefore, although we use the term "continual learning", we do not emphasize overcoming catastrophic forgetting as in most work of this field.

Practical industrial recommendation models follow the continual learning setting, where the models gradually extend acquired

knowledge by incrementally training with new data. However, continual learning for cross-domain recommendation has not been fully studied in both academia and industry. Recently proposed Conure [32] continually learns user representations from various tasks to better model the user preference and personality. SML [34] presents a sequential meta-learning method to retrain a recommender system by learning a network to connect past trained models to future models. However, SML is inefficient for industrial applications, and both Conure and SML are single-domain models so they can not handle cross-domain CTR prediction tasks.

## 3 METHODS

In this section, we first introduce the single-domain CTR prediction model in recommender systems of Taobao, then analyze the characteristics of cross-domain CTR prediction model at Taobao. Finally, we present CTNet which performs continual transfer learning for cross-domain CTR prediction at Taobao.

### 3.1 Single-Domain CTR Prediction Model at Taobao

Taobao is one of the largest e-commerce platforms in the world, with hundreds of million users and candidate items for recommendation. In recommender systems (RS) of Taobao, a multi-stage cascade ranking architecture is deployed online, which includes a matching stage (also called retrieval or candidate generation), a pre-ranking stage, and a ranking stage. We perform CTR prediction at the ranking stage, which is formally defined as:

**DEFINITION 1. CTR Prediction:** Given a user  $u$  and a target candidate item  $v$ , the objective is to estimate the probability  $p$  that  $u$  clicks on  $v$  with a prediction function:

$$h : (u, v) \rightarrow p \quad (1)$$

where the prediction function is optimized using the popular binary cross-entropy loss to measure the distance between  $p$  and  $y$ . Here  $y$  is the ground-truth probability distribution indicating whether the user clicks the item or not.

More specifically, each user-item interaction  $(u, v)$  is represented by a series of features such as *user features*, *item features*, *cross features*, and *context features*. These features can be summarized into *categorical features* like user ID, user gender, item ID, category ID, seller ID, etc, and *numerical features* like user's or item's historical statistics at Taobao. Features of each user-item interaction are transformed into dense embedding vectors via embedding lookup<sup>2</sup>.

In addition to the above-mentioned features, users' historical behavior sequences (including domain-specific behaviors and behaviors across different domains in Taobao) are also introduced, which has been demonstrated effective for CTR prediction [4, 36]. Concretely, Transformer-based target attention (TA) [4] is applied for short-term sequences, where a behavior sequence is firstly encoded by a single-layer Transformer encoder, then the encoded sequence is transformed by *target attention* (i.e., an attention mechanism using the target item as the query, and historical user behaviors as keys and values). For long-term user behaviors, target attention

<sup>2</sup>Categorical features are represented by one-hot encoding. Numerical features are pre-processed with discretization [9] to be transformed into one-hot encoding.

cannot be directly applied due to the strict constraint of inference time in real-world systems. Thus, we apply SIM (hard) [22] and ETA [2] to model the long behavior sequences. Both methods firstly use a well-designed retriever to select the top- $k$  items which are most similar to the target item from the long behavior sequence, then conduct target attention between the target item and the selected top- $k$  items. The main difference between SIM (hard) and ETA lies in the design of the retriever. To be specific, SIM (hard) retrieves items that share the same category with the target item, while ETA can retrieve items from distinct categories by utilizing the Locality-Sensitive-Hashing (LSH) approach to efficiently extract top- $k$  similar items.

The feature embeddings and attention layer outputs ( $A_{TA}$ ,  $A_{SIM}$ , and  $A_{ETA}$ ) are concatenated as the input of a multi-layer perceptron (MLP) with three hidden layers to get the prediction result:

$$h(u, v) = \sigma(\text{MLP}(E)) \quad (2)$$

$$E = \text{Concat}(E_u, E_v, E_{\text{cross}}, E_{\text{context}}, A_{TA}, A_{SIM}, A_{ETA}) \quad (3)$$

where  $\sigma$  denotes the sigmoid activation function.

### 3.2 Principle of Cross-Domain CTR Prediction Model Design at Taobao

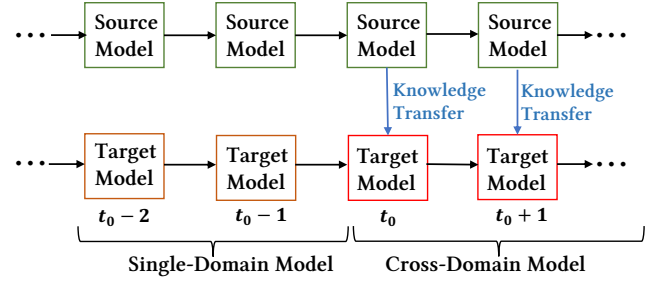
Taobao is a large-scale portal with hundreds of different-sized recommendation domains, where each domain has its own single-domain CTR prediction model. Here we summarize three key characteristics of recommendation domains in Taobao.

- Scales of different recommendation domains can vary significantly. A user can browse across domains, and an item can appear in multiple domains. Some domains can have tens or hundreds of times more user-item interactions than other domains.
- For the same user or item, its pattern can vary across different domains. For example, users tend to purchase urgently needed products in Taobao Live rather than in other recommendation domains. Items are shown with specially designed white-background pictures and stronger promotion information in JuHuaSuan and BaiYiBuTie.
- All the recommendation domain are time-evolving. So each single-domain CTR prediction model is trained under the continual learning setting to acquire new knowledge from incremental data. Therefore, all the single-domain CTR prediction model has been continually well-trained for a long time (years) so that these learned parameters contains useful knowledge that cannot be discarded.

In view of this, the cross-domain CTR prediction model needs to (1) transfer knowledge from the large domains to the small domains, (2) avoid conflicts between different domains' objective functions, and (3) efficiently learn the real-time knowledge from both source and target domains **without forgetting historical knowledge**. Figure 2 shows the usage of the cross-domain CTR prediction model.

### 3.3 CTNet: Continual Transfer Network

We define the cross-domain CTR prediction in Taobao as a Continual Transfer Learning (CTL) problem. In this subsection, we first formalize the CTL and then introduce our model CTNet to solve this problem.



**Figure 2: An illustration of deploying a cross-domain CTR prediction model in the continual learning setting. Before time step  $t_0$  we have a single-domain target model. At  $t_0$  we start to deploy a cross-domain model based on previously well-trained models.**

**Problem Formulation.** Cross-domain CTR prediction targets to transfer knowledge from the larger source domain to improve the CTR prediction performance on the smaller target domain, which is essentially a transfer learning problem. In many industrial platforms like Taobao, both the source domain and the target domain are synchronizing changing over time. So it is necessary to consider the transfer learning problem under a continual learning setting. In light of this, we propose continual transfer learning which is important but less explored in both academia and industry.

**DEFINITION 2. Continual Transfer Learning (CTL):** Given a time-evolving source domain  $\{\mathcal{D}_t^S\}_{t=1}^T$  and a time-evolving target domain  $\{\mathcal{D}_t\}_{t=1}^T$ , where  $t$  denotes the time step, continual transfer learning aims to improve future prediction performance on target domain  $\mathcal{D}_{T+1}$  using the historical and real-time knowledge from both the source domain and the target domain.

In industrial practice, a single-domain model is often initialized by its latest version and then trained with new data to achieve continual learning, which can be implemented by either *offline incremental learning* or *online learning*. Concretely, we collect new data during a time period between time step  $t$  and time step  $t+1$  to incrementally train the model. The time period can be any length of time, e.g., weekly, daily, or even minutely.

Our work involves three models including single-domain source domain model  $h_S(\cdot)$ , single-domain target domain model  $h(\cdot)$ , and the proposed cross-domain target model CTNet  $f(\cdot)$ . Specifically, the source domain model and the target domain model are continually trained respectively using the incremental source domain data and the incremental target domain data, so that they can timely adapt to the new environment.

**Model Architecture of CTNet.** As illustrated in Figure 3, CTNet follows a two-tower architecture, with a **source tower** and a **target tower**. The source tower has the same model architecture as source domain model, while the target tower has the same architecture as the target domain model. The source tower and the target tower are connected by several light-weighted **adapters**. At the time step  $t$ , all the parameters  $\Phi_t^S$  of the latest trained source domain model  $h_t^S(\cdot)$ , including embedding layers, attention layers and MLP layers, are built into the computation graph of  $f_t(\cdot)$  as the source tower and kept frozen during the training of CTNet. Distinct

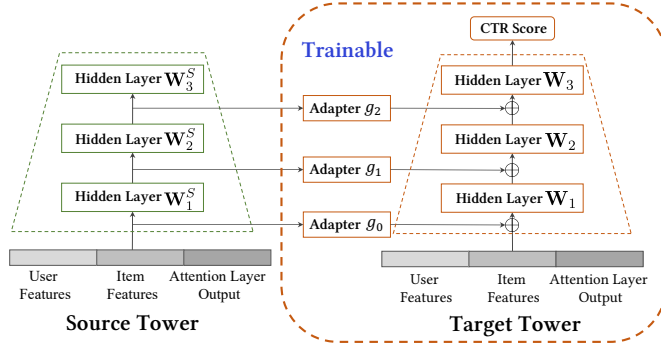


Figure 3: Architecture of the proposed CTNet model.

from existing pre-training & fine-tuning cross-domain CTR prediction methods, CTNet does not discard previously well-trained target domain model parameters but re-use all these parameters to initialize the target tower.

Light-weighted adapters parameterized by  $\Theta$  are adopted at each layer of the towers to learn the domain mapping from the source to the target. More clearly, an adapter  $g_l$  at the towers' (i.e., MLPs')  $l$ -th layer maps the hidden representation  $z_l^S$  in the source tower to be auxiliary knowledge for the target tower. Therefore, the hidden representation  $z_l$  in the target tower is computed by,

$$z_l = \psi \left( W_l z_{l-1} + g_l \left( z_l^S \right) \right) \quad (4)$$

where  $\psi$  denotes the activation function, and matrix  $W_l$  is trainable. Notably,  $z_0^S$  is the input  $E^S$  of the source tower.

Inspired by the success of gated units like GLU [3], we implement the adapter by a feature selection gate to adaptively control the information flow from the source tower to the target tower:

$$g_l \left( z_l^S \right) = U_l^1 z_l^S \odot \sigma \left( U_l^2 z_l^S \right) \quad (5)$$

where  $\sigma$  denotes the sigmoid activation function, and  $\odot$  denotes element-wise vector multiplication. Matrices  $U_l^1$  and  $U_l^2$  are trainable parameters.

For the proposed CTNet, only the target tower and the light-weighted adapters need to be optimized by SGD with the target domain CTR prediction objective function. At each time step, the source tower is updated with the latest source domain model parameters, and it kept fixed during the training of CTNet. The adapters can extract useful knowledge from the source tower to enhance the performance of target tower. Therefore, CTNet is able to utilize both the historically acquired and the latest knowledge from both the source domain and the target domain. More importantly, we just need incremental target domain data to efficiently train the model. Algorithm 1 shows the overall training process of CTNet.

#### Insights: Why Need Two Towers and How to Use the Adapters?

The network architectures of the source tower and the target tower are specially designed to be identical to the original source model and the original target model. This straightforward design can **make the transfer convenient** in industrial applications. In practice at Taobao, the single-domain source domain model  $h_t^S(\cdot)$  and target domain model  $h_t(\cdot)$  have been continually trained for a very long time, so they have captured users' long-term interests. In other

---

**Algorithm 1:** Training of CTNet. We assume CTNet is deployed at time step  $t_0$  and trained until  $t_0 + T$ .

---

**Input:** Target domain data  $\{\mathcal{D}_t\}_{t=t_0}^{t_0+T} = \{(X_t, Y_t)\}_{t=t_0}^{t_0+T}$ ;

Continually trained source models  $\{h_t^S\}_{t=t_0}^{t_0+T}$ ;

Previously trained single-domain target model  $h_{t_0}$ .

**Output:** target domain model  $f_{t_0+T}$ .

- 1 Randomly initialize adapter  $\Theta$  with small random values;
  - 2 Initialize source tower  $\Phi^S$  with  $h_{t_0}^S$ ;
  - 3 Initialize target tower  $\Phi$  with  $h_{t_0}$ .
  - 4 **for**  $t = t_0$  to  $t_0 + T$  **do**
  - 5     Source tower transfer: update source tower  $\Phi^S$  with  $h_t^S$ ;
  - 6     Update  $\Theta$  and  $\Phi$  by gradient descent:
  - 7      $\Theta_{t+1} \leftarrow \Theta_t - \alpha \nabla_{\Theta} \mathcal{L}(f_{\Theta, \Phi, \Phi^S}(X_t), Y_t)$ ;
  - 8      $\Phi_{t+1} \leftarrow \Phi_t - \alpha \nabla_{\Phi} \mathcal{L}(f_{\Theta, \Phi, \Phi^S}(X_t), Y_t)$ ;
  - 9 **end**
  - 10 **return** the model  $f_{t_0+T}$  parameterized by  $\Phi_{t_0+T}^S$ ,  $\Theta_{t_0+T}$ , and  $\Phi_{t_0+T}$ .
- 

words, the well-trained source domain model and target domain model are very valuable, and we need to fully utilize them. Benefiting from the source tower and target tower, CTNet can reuse all the parameters, including all the embedding layers, attention layers, and MLP layers of current source and target production models to preserve their strong prediction ability.

Notably, the current target domain production model is highly optimized with billions of data, so it converges into a local optimum. Therefore, significant parameter changes in the target tower can make the model deviate from this optimal solution thus hurting the CTR prediction performance. In view of this, **at the very beginning of continual transfer learning, we initialize the parameter  $\Theta$  in adapters with very small values**, so that the initial outputs of  $g_\Theta$  are close to zero. By doing this, we can regard the initial CTNet almost identical to that of the target domain model. Beside, the adapters integrate the source domain hidden representations and target domain hidden representations using addition operation instead of using concatenation operation. That is because we need to retain the dimension of hidden representations, such that we can reuse the well-trained target domain model parameters.

To sum up, the design of CTNet architecture seems simple and straightforward, but the methodology reveal useful practical knowledge that are less-addressed before — we should try to reuse previously well-learned target domain parameters as many as possible when conducting transfer learning. This is often not a issue for the experiments on offline benchmarks, while it is the key reason why CTNet is successfully deployed online at Taobao, and largely outperforms most of the other state-of-the-art cross-domain CTR prediction methods.

## 4 EXPERIMENTS

In this section, we conduct experiments to analyze the following research questions (RQs): (1) How does CTNet perform compared with existing cross-domain CTR prediction and single-domain methods? (2) Why is it necessary to continually transfer knowledge? (3)



Why are pre-training & fine-tuning methods unsuitable for continual transfer learning? and (4) Can CTNet help address the cold-start problem?

#### 4.1 Datasets

In this paper we address the real-world continual cross-domain CTR prediction problem at Taobao. The presented problem has a series of characteristics related to industrial applications. Specifically, we assume (1) the previously deployed online models are well-trained for a long time (e.g. years) preserving very long-term knowledge. (2) users and items may share in different domains (3) very large scale of (e.g. billions of) sequential streaming data is used for training. In fact, models of industrial systems from large portals like Taobao, Google, Amazon, TikTok, etc. usually satisfy these criteria. Unfortunately, there does not exist a suitable public benchmark for simulating the situation for the continual cross-domain CTR prediction problem in Taobao. So we can only evaluate our approach on Alibaba production data and online experiments on Taobao mobile app. Three different-sized recommendation domains — domain A, B, and C are used for evaluating transferring performances from domain A to B, and A to C. Domain A, the largest recommendation domain at Taobao, has about 60 times more user-item interaction data than domain B, and about 200 times more data than domain C. Each domain shares some users and items, whereas domain A has covered most users and items in Taobao. The numbers of different users and items in domains B and C are much smaller. We collect and sample traffic logs of domains A, B, and C to get user-item interaction data for 31 days. Data of the last day is used for testing and data of the other days is used for training. We organize each training dataset into periods according to the time of interaction. Each period lasts for 6 days so each dataset is split into 5 periods. A user-item interaction sample indicates an impression interaction of a user to an item that has been viewed by the user. If the user clicked the item and then the sample is labeled as positive. Otherwise, the sample is labeled as negative. The sizes of training data in domain A, domain B, and domain C are about 150 billion, 2 billion, and 1 billion, respectively. We use the down-sampling technique to keep all positive samples and a portion of the negative samples of domain A, due to the extremely large scale data. Note that although domains B and C are smaller than A, there are still a huge amount of user-item interaction data, with hundreds of millions of active users.

#### 4.2 Experimental Details

This work focuses on the application of CTL for industrial recommender systems. Therefore, experiments are conducted based on our online production model, ensuring the practical applications of the experiments. We reuse as many parameters as possible from our production models for all the methods in experiments to achieve good performance. The production models are highly optimized and have been continually trained for 2 years so that an absolute value gain of 0.1% on AUC/GAUC is considered a significant improvement. For all the methods, we set the mini-batch size of 512, and the number of hidden units of the 3-layer MLP are [1024, 512, 256]. Multi-head target attention with 8 heads and 256 hidden units is used in user behavior modeling. The sub-sequence

lengths of SIM and ETA are set to be 64. In addition, leaky-ReLU activation function  $\psi$ , batch normalization, AdaGrad optimizer with an initial learning rate  $\alpha = 0.01$  are adopted. For a few features that are only available in the source domain, we set the feature values to be zero or empty. All the experiments are conducted on a large-scale distributed machine learning training platform.

#### 4.3 Evaluation Metrics

Following previous work in CTR prediction, we apply AUC and Group AUC (GAUC) [38] for evaluation. GAUC is user group AUC weighted by the impressions of each user. It measures the goodness of intra-user order and it is shown to be consistent with online performances [36]. GAUC is calculated as follows:

$$\text{GAUC} = \frac{\sum_{i=1}^n \# \text{impressions}_i \times \text{AUC}_i}{\sum_{i=1}^n \# \text{impressions}_i} \quad (6)$$

where  $n$  is the number of users,  $\# \text{impressions}_i$  and  $\text{AUC}_i$  are the numbers of impressions and AUC of the  $i$ -th user, respectively.

#### 4.4 Compared Methods

Various methods are adopted for comparison, including pre-training and fine-tuning Cross-Domain CTR prediction methods:

- **Pre-train & Fine-tune (ID Embeddings):** target domain model that uses pre-trained ID embeddings<sup>3</sup> from the source domain model and fine-tunes on the target dataset.
- **Pre-train & Fine-tune (All Embeddings):** target domain model that uses all the pre-trained embeddings of shared features across two domains from the source domain model and fine-tunes on the target domain dataset.
- **Pre-train & Fine-tune (All Parameters):** fine-tune the entire source domain model on the target dataset. For features only available in the source domain, we set the feature values to be zero or empty. For fair comparisons, we add the features only available in the target domain so features are identical to target domain models.
- **Dual Embedding:** ID embeddings from the latest source domain model are added into the target domain model as extra embeddings. So that there are two item IDs, user IDs, etc. in the model. All parameters and features are updated in training.

And joint learning cross-Domain CTR prediction methods:

- **MLP++**[12]: A multi-task model that shares the parameters of the user embeddings only. We treat each domain as a task in multi-task learning.
- **Share Bottom**[18]: A multi-task model that shares the parameters of the bottom layers (i.e., the embedding layer of shared features in source and target domains).
- **PLE**[25]: A multi-task Mixture-of-Expert (MoE) model with gating networks, which improves based on MMoE [18].
- **MiNet** [21]: Mixed Interest Network (MiNet) for cross-domain CTR prediction. It models three types of user interest from the source and target domain.

<sup>3</sup>User ID, item category ID, item ID, and item seller ID. These ID embeddings account for more than 97% of parameters in the CTR prediction network.

- **DDTCDR** [16]: Combining deep dual transfer learning mechanism and latent embedding approach for cross-domain CTR prediction.
- **DASL** [15]: Dual attentive sequential learning for cross-domain CTR prediction with dual embedding and dual attention, which is the state-of-the-art method for cross-domain CTR prediction.

We also report results of several single-domain models:

- **Base (w/o transfer)**: our single-domain target domain production model described in Section 3.1.
- **Source Domain Model**: directly using the predictions from the source domain model without fine-tuning.
- **DIN** [36]: deep interest network, a widely adopted baseline model in industrial recommender systems. Our base production model has several more model components compared with DIN, including SIM and ETA.

For fair comparisons, we implement these methods with the same model architecture, hyper-parameters, training procedure, and features with CTNet.

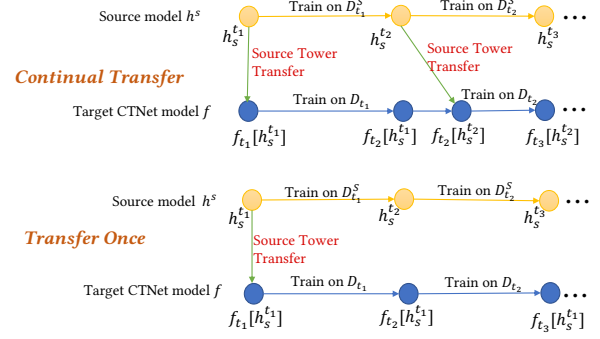
#### 4.5 Overall Results (RQ1)

We have conducted detailed analysis that CTNet is **much more efficient in training** than existing cross-domain CTR prediction methods in Section 1. Then we will find CTNet also **outperforms these methods in real-world datasets** according to the overall results shown in Table 2. We summarize the following conclusions:

(1) **CTNet significantly outperforms all the baselines** including joint learning cross-domain CTR prediction, pre-training & fine-tuning cross-domain CTR prediction, and single-domain models in two transferring tasks.

(2) In comparison with pre-training & fine-tuning methods, CTNet is not only more efficient and suitable in the continual transfer learning setting but also largely surpasses them in performance gains. This is partly due to that these methods replace parts or all model parameters with source model parameters, thus needing to re-learn the historically acquired knowledge in the original target model. The performance of Pre-train & Fine-tune (All Parameters) is even worse than our base production model due to large domain differences. In contrast, CTNet re-uses all the available parameters from the single-domain target model for **minimum information loss during transfer**. Besides, existing methods like Pre-train & Fine-tune (ID Embeddings and All Embeddings) only utilize limited knowledge from partial parameters from the source domain model, but CTNet uses all the layers, especially **deep intermediate representations that accumulate higher-level prediction abilities** in MLP layers of the source domain model.

(3) In addition to that joint learning cross-domain CTR prediction is inefficient for training as stated in Section 1, these methods do not perform well in real-world datasets. On the one hand, these approaches transfer knowledge mainly by auxiliary losses to update low-level features, so the transfer process is not as direct as other methods. And extra objectives from the source domain may not guarantee the optimal performances of the target due to conflicts of different objectives [1, 30]. On the other hand, these methods only take advantage of the limited 30-days source domain training data



**Figure 4: Illustration of the comparison of continual transfer and transfer once. Model at time  $t$  where the source tower is updated by source model at time  $\tau$  is denoted as  $f^t[h^{\tau}_s]$ .**

without fully utilizing the power of pre-trained source domain models that contain historically acquired knowledge learned for years. In contrast, CTNet utilizes continual long-term knowledge from all the historical data of source and target domain. It re-uses all the available parameters and dense hidden representations from source models. Both low-level features from embedding layers and high-level representations from MLP layers are leveraged for **maximum information gain from the source domain**, thus achieving superior results.

(4) CTNet shows more significant improvements in transferring knowledge from domain A to C, than domain A to B. Firstly, domain C has less training data than domain B so the data sparsity issue is more severe. Secondly, almost all items in domain B are old items, which limits the performance gains from transfer learning. However, domain C has about 40% of newer items that have not accumulated enough user-item interaction data compared with older items. In other words, domain C has a more serious cold-start item recommendation problem. CTNet shows prominent ability on the recommendation for these new items (detailed analysis in Section 4.6), thus achieving high overall performance gains.

(5) We conduct ablation study on the gated linear unit (GLU) of the adapters. **CTNet (w/o GLU)** replaces GLU adapters with simpler linear adapters. Experiments shows that GLU performs slightly better than linear layers, verifying the effectiveness of feature selection of gated units.

#### 4.6 Detailed Experimental Analysis

This section analyzes the necessity of continual transfer; why pre-training & fine-tuning methods are impractical in industrial applications, and the ability of CTNet for cold-start recommendation.

**Continually Transferring Knowledge Is Better Than Transferring Only Once (RQ2)** We then show the performance of CTNet with two different settings: continual transfer and transfer once. Figure 4 illustrates the two different strategies, where the top is *continually transferring* the knowledge from the source to the target (proposed in this paper, as stated in Algorithm 1), and the bottom is transferring knowledge *only once* which is commonly adopted in traditional transfer learning methods.

**Table 2: Offline model evaluation with source domain A, target domain B and C.**

Model	Domain A to B		Domain A to C	
	AUC	GAUC	AUC	GAUC
<b>Base (w/o transfer)</b>	0.7404	0.6788	0.7104	0.6677
<b>Source Domain Model</b>	0.6640	0.6200	0.5783	0.5771
<b>DIN</b>	0.7238	0.6593	0.7037	0.6537
<b>Pre-train &amp; Fine-tune (ID Embeddings)</b>	0.7439	0.6828	0.7190	0.6769
<b>Pre-train &amp; Fine-tune (All Embeddings)</b>	0.7438	0.6827	0.7182	0.6767
<b>Pre-train &amp; Fine-tune (All Parameters)</b>	0.7368	0.6739	0.6727	0.6352
<b>Dual Embedding</b>	0.7402	0.6787	0.7137	0.6693
<b>MLP++</b>	0.7405	0.6764	0.7116	0.6710
<b>Share Bottom</b>	0.7417	0.6777	0.7119	0.6726
<b>PLE</b>	0.7395	0.6749	0.7103	0.6679
<b>MiNet</b>	0.7411	0.6765	0.7129	0.6712
<b>DDTCDR</b>	0.7408	0.6768	0.7118	0.6709
<b>DASL</b>	0.7406	0.6763	0.7132	0.6714
<b>CTNet (w/o GLU)</b>	0.7465	0.6877	0.7432	0.7023
<b>CTNet</b>	<b>0.7474</b>	<b>0.6888</b>	<b>0.7451</b>	<b>0.7040</b>

**Table 3: Results of the comparison of continual transfer and transfer once from domain A to B. Numbers in brackets indicate the AUC/GAUC gain compared with base model. Data of the next day of the time period is used as testing sets.**

Time	Model	AUC	GAUC
$t_2$	<b>Base(w/o transfer)</b>	0.7483	0.6931
	<b>CTNet</b>	0.7548(+0.0065)	0.7022(+0.0091)
$t_3$	<b>Base(w/o transfer)</b>	0.7385	0.6728
	<b>CTNet(transfer once)</b>	0.7440(+0.0055)	0.6822(+0.0094)
	<b>CTNet</b>	0.7449(+0.0064)	0.6829(+0.0101)
$t_4$	<b>Base(w/o transfer)</b>	0.7416	0.6788
	<b>CTNet(transfer once)</b>	0.7462(+0.0046)	0.6856(+0.0068)
	<b>CTNet</b>	0.7486(+0.0070)	0.6882(+0.0094)
$t_5$	<b>Base(w/o transfer)</b>	0.7424	0.6782
	<b>CTNet(transfer once)</b>	0.7457(+0.0033)	0.6862(+0.0080)
	<b>CTNet</b>	0.7498(+0.0075)	0.6877(+0.0095)
$t_6$	<b>Base(w/o transfer)</b>	0.7404	0.6788
	<b>CTNet(transfer once)</b>	0.7436(+0.0032)	0.6861(+0.0073)
	<b>CTNet</b>	0.7474(+0.0070)	0.6888(+0.0100)

As summarized in Table 3, the results of continual transfer are consistently better than transfer once. Performance gains of transfer once drop with time-evolving, but gains from continual transfer sustain. Therefore, if transfer learning is conducted only once, the model cannot quickly adapt to the changing environment. Continual transfer is essential for cross-domain CTR prediction to consistently acquire new knowledge from the source domain.

**Pre-training and Fine-tuning Is Not Suitable for CTL (RQ3)**  
Experimental results in Table 4 shows why popular pre-training & fine-tuning methods are impractical for industrial RS. We analyze the results of transferring knowledge from domain A to domain B with Pre-train & Fine-tune (All Parameters) strategy with different sizes of training data. The data size is represented by the duration of fine-tuning data (about 100M samples per day).

**Table 4: Pre-train & Fine-tune (All Parameters) results with different sizes of fine-tuning data.**

	AUC	GAUC
<b>Base (w/o transfer)</b>	0.7404	0.6788
<b>Pre-train &amp; Fine-tune (7 days)</b>	0.7291	0.6666
<b>Pre-train &amp; Fine-tune (30 days)</b>	0.7368	0.6739
<b>Pre-train &amp; Fine-tune (45 days)</b>	0.7397	0.6781
<b>Pre-train &amp; Fine-tune (120 days)</b>	0.7401	0.6786
<b>CTNet</b>	0.7474	0.6888

As is illustrated in Table 4, the performance of pre-training & fine-tuning is much worse than the base model if the size of fine-tuning data is relatively small. A very large amount of data (over 10,000M samples) is needed to catch up with the performance of the base production model. The main reason is that in pre-training and fine-tuning, existing well-trained target domain parameters which store very long-term domain knowledge are abandoned and replaced by source domain parameters. So a large amount of target domain data is needed to re-learn the knowledge to get good fine-tuning results. This indicates the impracticality of applying pre-training & fine-tuning to CTL. It is extremely costly that every time the pre-trained source domain model is updated with new data, the model needs to be fine-tuned with large-size target domain data. On the contrary, if the transfer process is conducted less frequently for efficiency, then the performance gains of transfer will drop with time-evolving (as stated in the above analysis of continual transfer and transfer once). However, the proposed CTNet keeps all the previously learned target domain parameters to preserve historically acquired long-term domain-specific knowledge, thus only needing incremental data at the current time step for training the model. So it can conveniently and continually transfer knowledge by leveraging the latest continually pre-trained source domain model every time it is updated.



**Table 5: Cold-start recommendation results. AUC gains are shown in absolute values, and CTR gains are shown in relative improvement.**

Type	AUC Gain	GAUC Gain	CTR Gain
<b>New Users</b>	+0.0114	+0.0150	+2.6%
<b>Old Users</b>	+0.0052	+0.0058	+2.5%
<b>New Items</b>	+0.0102	N/A	+26.3%
<b>Old Items</b>	+0.0124	N/A	+1.9%

**CTNet Can Be Applied to Cold-Start Recommendation (RQ4)**

We evaluate CTNet on the transferring performance from domain A to B on cold-start recommendation, for both cold-start users and items of domain B. Most of the new users and new items of domain B have been in domain A before. Table 5 shows offline and online performance gains of different types of users and items. Compared with the previous online production model (Base(w/o transfer)), CTNet achieves higher AUC/GAUC/CTR gains for new users compared with old users. As for new items, offline metrics for new items and old items are not comparable as AUC results are evaluated on different groups of items. But we observe a significant 26.3% CTR gain in online A/B testing, demonstrating the **promising ability of CTNet for cold-start item recommendation**.

**4.7 Productionizing CTNet at Taobao**

We productionize CTNet on two large-scale recommender systems in Taobao (domain B and C). When compared with our previous production models, CTNet yielded 1.0% GAUC improvement in domain B and 3.6% GAUC improvement in domain C (as is shown in Table 2). Although domain B and C are smaller than domain A, they still have very large traffic with hundreds of millions of active users. For these highly-optimized production models, a gain of 0.1% on GAUC is considered a significant improvement. So the improvements of CTNet on production models are highly remarkable. We also observed consistently significant online A/B testing performance gains with 2.5% improvement on CTR and 7.7% improvements on Gross Merchandise Volume (GMV) in domain B, and 12.3% improvement on CTR and 31.9% improvements on GMV in domain C, which greatly increase total profit considering the large traffic of these domains.

To reduce the online inference cost, attention layers of source tower and target tower with the same user behavior sequences (i.e. SIM and ETA<sup>4</sup>) are shared during training and inference. This approach does not lead to differences in model performance but reduces most extra computation costs from CTNet. With the same online computation resources, we do not observe any changes in online response time comparing CTNet to previous production models. Since December 2021, CTNet has been deployed online at Taobao and consistently bring significant improvements.

**5 CONCLUSIONS AND DISCUSSIONS**

In this paper, we addresses the problem of cross-domain CTR prediction at Taobao. We present a novel continual transfer learning

<sup>4</sup>SIM and ETA utilize user behavior sequences of all domains in Taobao. These two modules consume the most online computing cost in our production model. We replace SIM and ETA components in the target tower with that in the source tower.

(CTL) task: transferring knowledge where both source and target domain are time-evolving. The continual transfer network (CTNet) is proposed to effectively and efficiently transfer knowledge. CTNet connects continually pre-trained source domain models to the target model to improve the target performance. It takes advantage of all the well-trained parameters from source and target domain models.

We believe that the proposed CTL is not only applicable to recommendation, search, and advertising but will also have broader applications in other real-world scenarios with time-evolving domains, such as financial modeling, dialogue systems, autonomous driving, etc.

**REFERENCES**

- [1] Lei Chen, Fajie Yuan, Jiayi Yang, Xiangnan He, Chengming Li, and Min Yang. 2021. User-specific Adaptive Fine-tuning for Cross-domain Recommendations. *arXiv preprint arXiv:2106.07864* (2021).
- [2] Qiwei Chen, Changhua Pei, Shanshan Lv, Chao Li, Junfeng Ge, and Wenwu Ou. 2021. End-to-End User Behavior Retrieval in Click-Through Rate Prediction Model. *CoRR abs/2108.04468* (2021). [arXiv:2108.04468](https://arxiv.org/abs/2108.04468)
- [3] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. 2016. Language Modeling with Gated Convolutional Networks. (2016).
- [4] Chen et al. 2020. Behavior sequence transformer for e-commerce recommendation in alibaba. In *International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (DLP-KDD)*.
- [5] Heng-Tze Cheng et al. 2016. Wide & Deep Learning for Recommender Systems. (2016). [arXiv:1606.07792](https://arxiv.org/abs/1606.07792)
- [6] J. Kirkpatrick et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)* (2017).
- [7] Utku Evci, Vincent Dumoulin, Hugo Larochelle, and Michael C. Mozer. 2022. Head2Toe: Utilizing Intermediate Representations for Better Transfer Learning. *CoRR abs/2201.03529* (2022). <https://arxiv.org/abs/2201.03529>
- [8] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. 2019. Dlow: Domain flow for adaptation and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [9] H. Guo, B. Chen, R. Tang, Z. Li, and X. He. 2021. AutoDis: Automatic Discretization for Embedding Numerical Features in CTR Prediction. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- [10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *International Joint Conference on Artificial Intelligence (IJCAI)*.
- [11] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morroni, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *International Conference on Machine Learning (ICML)*.
- [12] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the ACM International Conference on Information & Knowledge Management (CIKM)*.
- [13] Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2021. Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora. *arXiv preprint arXiv:2110.08534* (2021).
- [14] M. Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, and A. et al. Leonardi. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).
- [15] Pan Li, Zhichao Jiang, Maofei Que, Yao Hu, and Alexander Tuzhilin. 2021. Dual Attentive Sequential Learning for Cross-Domain Click-Through Rate Prediction. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*.
- [16] Pan Li and Alexander Tuzhilin. 2020. Dtdcd: Deep dual transfer cross domain recommendation. In *Proceedings of the International Conference on Web Search and Data Mining (ICDM)*.
- [17] Hong Liu, Mingsheng Long, Jianmin Wang, and Yu Wang. 2020. Learning to Adapt to Evolving Domains. In *NeurIPS*.
- [18] 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 (KDD)*.
- [19] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019.  $\pi$ -Net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.
- [20] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, and Maarten De Rijke. 2019. pi-Net: A Parallel Information-sharing Network for Shared-account Cross-domain

- Sequential Recommendations. *ACM* (2019).
- [21] Wentao Ouyang, Xiuwu Zhang, Lei Zhao, Jinmei Luo, Yu Zhang, Heng Zou, Zhaojie Liu, and Yanlong Du. 2020. MiNet: Mixed Interest Network for Cross-Domain Click-Through Rate Prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM)*.
  - [22] Pi Qi, Xiaoqiang Zhu, Guorui Zhou, Yujing Zhang, and Kun Gai. 2020. Search-based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction. In *Proceedings of the ACM International Conference on Information & Knowledge Management (CIKM)*.
  - [23] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive Neural Networks. (2016). arXiv:1606.04671
  - [24] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34.
  - [25] 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 '20: Fourteenth ACM Conference on Recommender Systems*.
  - [26] A. Taufique, S. Jahan, and A. Savakis. 2021. ConDA: Continual Unsupervised Domain Adaptation. *arXiv preprint arXiv:2103.11056* (2021).
  - [27] Hao Wang, Hao He, and Dina Katabi. 2020. Continuously indexed domain adaptation. *arXiv preprint arXiv:2007.01807* (2020).
  - [28] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. *CoRR* abs/1708.05123 (2017). arXiv:1708.05123
  - [29] Jun Wu and Jingrui He. 2021. Continuous Transfer Learning. <https://openreview.net/forum?id=dJbf5SqBFrM>
  - [30] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient Surgery for Multi-Task Learning. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*.
  - [31] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.
  - [32] F. Yuan, G. Zhang, A. Karatzoglou, X. He, J. Jose, B. Kong, and Y. Li. 2021. One Person, One Model, One World: Learning Continual User Representation without Forgetting. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.
  - [33] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021. Deep Learning for Click-Through Rate Estimation. (2021).
  - [34] Yang Zhang, Fuli Feng, Chenxu Wang, Xiangnan He, Meng Wang, Yan Li, and Yongdong Zhang. 2020. How to Retrain Recommender System?: A Sequential Meta-Learning Method. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.
  - [35] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2018. Deep Interest Evolution Network for Click-Through Rate Prediction. In *AAAI Conference on Artificial Intelligence (AAAI)*.
  - [36] Guorui Zhou, Chengru Song, Xiaoqiang Zhu, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2017. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.
  - [37] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. DTCDR: A Framework for Dual-Target Cross-Domain Recommendation. In *ACM International Conference on Information and Knowledge Management*.
  - [38] 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 *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM.
  - [39] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020).