



# Masked Multi-Domain Network: Multi-Type and Multi-Scenario Conversion Rate Prediction with a Single Model

Wentao Ouyang  
Xiuwu Zhang  
Alibaba Group  
maiwei.oywt@alibaba-inc.com  
xiuwu.zxw@alibaba-inc.com

Chaofeng Guo  
Shukai Ren  
Alibaba Group  
chaofeng.gcf@alibaba-inc.com  
shukai.rsk@alibaba-inc.com

Yupei Sui  
Kun Zhang  
Alibaba Group  
yupei.syp@alibaba-inc.com  
jerry.zk@alibaba-inc.com

Jinmei Luo  
Yunfeng Chen  
Alibaba Group  
cathy.jm@alibaba-inc.com  
chenyunfeng@alibaba-inc.com

Dongbo Xu  
Xiangzheng Liu  
Alibaba Group  
dongbo.xdb@alibaba-inc.com  
xiangzheng.lxz@alibaba-inc.com

Yanlong Du  
Alibaba Group  
yanlong.dyl@alibaba-inc.com

## ABSTRACT

In real-world advertising systems, conversions have different types in nature and ads can be shown in different display scenarios, both of which highly impact the actual conversion rate (CVR). This results in the multi-type and multi-scenario CVR prediction problem. A desired model for this problem should satisfy the following requirements: 1) Accuracy: the model should achieve fine-grained accuracy with respect to any conversion type in any display scenario. 2) Scalability: the model parameter size should be affordable. 3) Convenience: the model should not require a large amount of effort in data partitioning, subset processing and separate storage.

Existing approaches cannot simultaneously satisfy these requirements. For example, building a separate model for each (conversion type, display scenario) pair is neither scalable nor convenient. Building a unified model trained on all the data with conversion type and display scenario included as two features is not accurate enough.

In this paper, we propose the Masked Multi-domain Network (MMN) to solve this problem. To achieve the accuracy requirement, we model domain-specific parameters and propose a dynamically weighted loss to account for the loss scale imbalance issue within each mini-batch. To achieve the scalability requirement, we propose a parameter sharing and composition strategy to reduce model parameters from a product space to a sum space. To achieve the convenience requirement, we propose an auto-masking strategy which can take mixed data from all the domains as input. It avoids the overhead caused by data partitioning, individual processing and separate storage. Both offline and online experimental results validate the superiority of MMN for multi-type and multi-scenario CVR prediction. MMN is now the serving model for real-time CVR prediction in UC Toutiao.

## CCS CONCEPTS

• Information systems → Online advertising;

## KEYWORDS

Online advertising; CVR prediction; Multiple conversion types; Multiple display scenarios

## ACM Reference Format:

Wentao Ouyang, Xiuwu Zhang, Chaofeng Guo, Shukai Ren, Yupei Sui, Kun Zhang, Jinmei Luo, Yunfeng Chen, Dongbo Xu, Xiangzheng Liu, and Yanlong Du. 2023. Masked Multi-Domain Network: Multi-Type and Multi-Scenario Conversion Rate Prediction with a Single Model. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*, October 21–25, 2023, Birmingham, United Kingdom. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3583780.3614697>

## 1 INTRODUCTION

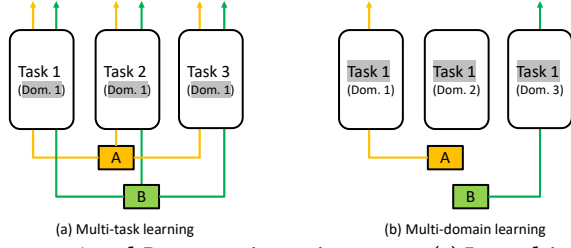
Conversion rate (CVR) prediction [6, 13, 15] is an essential task in online advertising systems. In optimized Cost-Per-Click (oCPC) and Cost-Per-Action (CPA) advertising, advertisers seek to maximize conversions for a given budget [18, 22, 32]. The predicted CVR impacts both the ad ranking strategy and the ad charging model.

Conversions have different types in nature (e.g., *purchase* a product vs. *sign up* an account) and ads can be shown in different display scenarios (e.g. shown as a *banner* on a page vs. interleaved with news in the *news channel*), both of which highly impact the actual CVR. As an evidence, the ground-truth CVRs vary significantly over different conversion types (e.g., from 6.3% to 18.9%) and over different display scenarios (e.g., from 0.9% to 14.9%) in the Criteo dataset (Table 1). This results in the multi-type and multi-scenario CVR prediction problem.

A desired model for this problem should satisfy the following requirements: 1) Accuracy: the model should achieve fine-grained accuracy with respect to any conversion type in any display scenario. 2) Scalability: the model parameter size should be affordable. 3) Convenience: the model should not require a large amount of effort in data partitioning, subset processing and separate storage.

One approach to solve this problem is to build a separate model for each (conversion type, display scenario) pair. However, it is neither scalable nor convenient. Another approach is to build a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
CIKM '23, October 21–25, 2023, Birmingham, United Kingdom  
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-0124-5/23/10...\$15.00  
<https://doi.org/10.1145/3583780.3614697>



**Figure 1:** A and B are two input instances. (a) In multi-task learning, domains are the same and tasks are different. One input impacts the model parameters of all tasks. (b) In multi-domain learning, domains are different and tasks are the same. One input impacts the parameters for that domain.

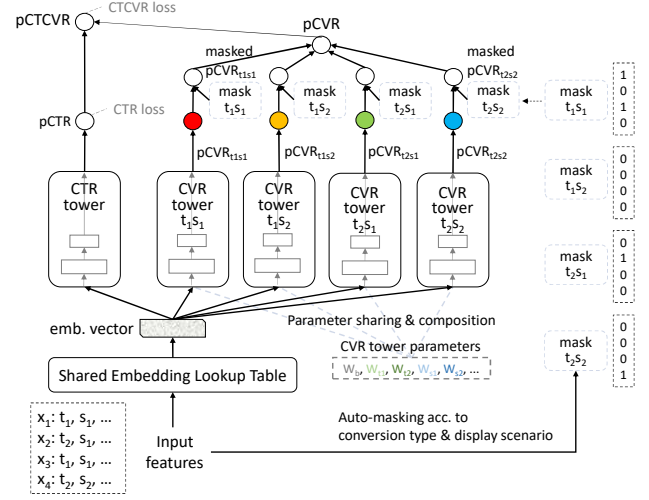
unified model trained on all the data, with conversion type and display scenario included as two feature fields. However, it fails to capture different natures of various conversion types and display scenarios, and is thus not accurate enough.

In this paper, we formulate the multi-type and multi-scenario CVR prediction problem as a multi-domain learning problem. We would like to illustrate the differences between multi-task learning [5] and multi-domain learning [9] in Figure 1. In multi-task learning, domains are the same and tasks are different. For example, given a video, simultaneously estimating what rating a user will give and how long the user will spend on the video is a multi-task learning problem [31]. In multi-domain learning, domains are different and tasks are the same. For example, estimating the CVRs of two instances of conversion type “purchase” and “sign up” by the same model is a multi-domain learning problem. Moreover, in multi-task learning, one input impacts the model parameters of all tasks; while in multi-domain learning, one input only impacts the model parameters corresponding to that domain. Because of such differences, existing multi-task learning models such as Shared-Bottom, OMoE, MMoE, CGC and PLE [16, 17, 27, 31] cannot well address our problem. MT-FwFM [22], STAR [25] and ADIN [11] are more relevant state-of-the-art multi-domain models. However, as they are not designed for our problem, they also cannot satisfy the aforementioned three requirements.

Given these limitations, we propose the Masked Multi-domain Network (MMN) in this paper. We will show how we design various strategies to make MMN simultaneously satisfy the accuracy, scalability and convenience requirements.

The main contributions of this work are summarized as follows:

- (1) We propose the MMN model to address the multi-type and multi-scenario CVR prediction problem in a real-world advertising system.
- (2) In order to achieve the accuracy requirement, we model domain-specific parameters and we propose a dynamically weighted loss to account for the loss scale imbalance issue within each mini-batch.
- (3) In order to achieve the scalability requirement, we propose a parameter sharing and composition strategy to reduce model parameters from a product space to a sum space.
- (4) In order to achieve the convenience requirement, we propose an auto-masking strategy which can take mixed data from all the domains as input. It avoids the overhead caused by data partitioning, individual processing and separate storage.



**Figure 2:** Masked multi-domain network. For simplicity, all the CTR and CVR prediction towers use the multi-layer perceptron consisting of several fully connected layers.

- (5) We conduct both offline and online experiments to test the performance of MMN and several state-of-the-art methods for multi-type and multi-scenario CVR prediction. We make the source code publicly available<sup>1</sup>.

## 2 MASKED MULTI-DOMAIN NETWORK

### 2.1 Multi-Type and Multi-Scenario CVR Prediction

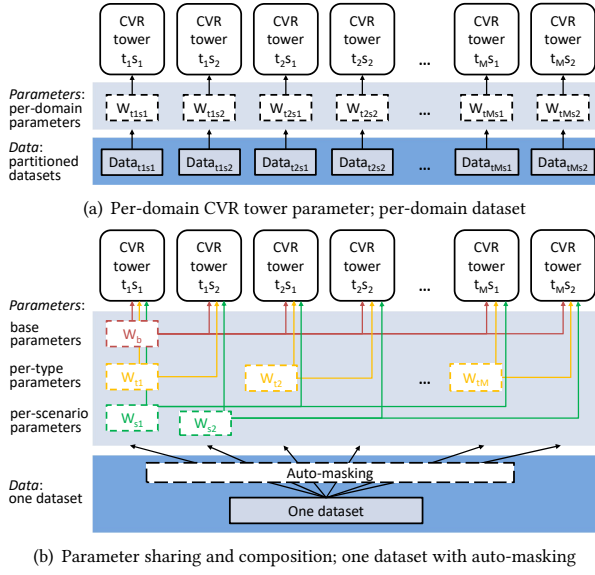
In omnimedia advertising platforms, different conversion types (e.g., purchase a product, sign up an account or fill out an online form) and different display scenarios (e.g., shown as a banner, interleaved with news or interleaved with micro-videos) highly impact the actual CVR. In the Criteo dataset (Table 1), the ground-truth CVRs vary significantly over different conversion types (e.g., from 6.3% to 18.9%) and over different display scenarios (e.g., from 0.9% to 14.9%). In our news feed advertising dataset (Table 1), the corresponding values vary from 0.05% to 27.6% and from 1.6% to 4.1% respectively.

Denote the input instance as  $x$ , which contains multiple features, such as user features, ad features, context features and cross features. Denote the conversion type of the ad as  $f_t(x)$  and the display scenario of the ad as  $f_s(x)$ . Further denote the click label as  $y \in \{0, 1\}$  and the conversion label as  $z \in \{0, 1\}$ . Mathematically, the multi-type and multi-scenario CVR prediction problem is to estimate  $p(z = 1 | y = 1, x, f_t(x), f_s(x))$ . It is a feature-rich problem.

### 2.2 Modeling Adaptive Parameters

Figure 2 illustrates the structure of the proposed MMN model. It consists a CTR tower and multiple CVR towers. The inclusion of a CTR tower is to utilize the available data in the entire sample space (i.e., not only *click* data but also *impression* data) [18]. For simplicity, all the CTR and CVR towers use the multi-layer perceptron (MLP) consisting of several fully connected layers. We treat each (conversion type, display scenario) pair as a domain and create a CVR tower

<sup>1</sup><https://github.com/oywtece/mmn>



**Figure 3: (a) Per-domain CVR tower parameter; per-domain dataset. (b) Parameter sharing and composition; one dataset with auto-masking.**

for each domain. But we use a parameter sharing and composition strategy to reduce the number of domain-specific parameters from a product space to a sum space. At the same time, these parameters can well capture each domain's specific properties.

Given an input instance  $x$ , we denote the concatenated feature embedding vector after embedding lookup as  $\mathbf{x}$ . We then generate the CVR prediction logit with ReLU activations [20] as

$$h = MLP(\mathbf{x}; \theta(f_t(x), f_s(x))), \quad (1)$$

where  $\theta$  is the set of model parameters,  $f_t(x)$  is the conversion type and  $f_s(x)$  is the display scenario of the input instance  $x$ .

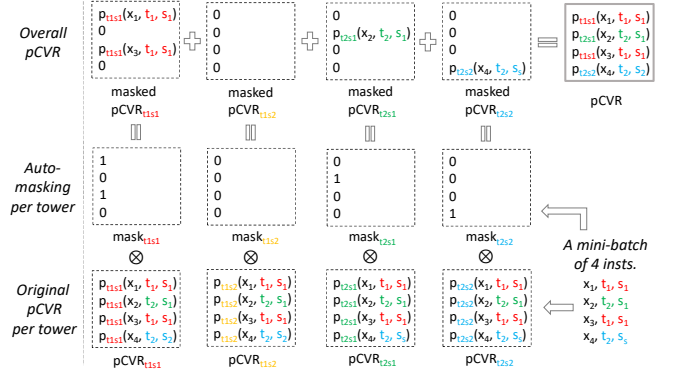
One possible way to model  $\theta(f_t(x), f_s(x))$  is to create a separate set of parameters for each (conversion type, display scenario) domain to make the model adaptive, as illustrated in Figure 3(a). Denote the number of conversion types as  $N_t$  and the number of display scenarios as  $N_s$ . It will result in  $N_t \times N_s$  sets of parameters, which makes the model too large to be trained.

In order to 1) capture commonalities across different domains, 2) reflect specific natures of different conversion types and display scenarios and 3) reduce the number of parameters, we design a parameter sharing and composition strategy as follows (illustrated in Figure 3(b)). In particular, we design

$$\theta(f_t(x), f_s(x)) = \{\mathbf{W}_b^l\}_{l=1}^L + \{\mathbf{W}_{f_t(x)}^l\}_{l=1}^L + \{\mathbf{W}_{f_s(x)}^l\}_{l=1}^L.$$

$\{\mathbf{W}_b^l\}_l$  is a set of common parameters that is shared by all the domains, in order to capture domain commonalities.  $L$  is the number of fully connected layers.  $\{\mathbf{W}_{f_t(x)}^l\}_l$  is a set of conversion type-specific parameters for  $f_t(x)$  and  $\{\mathbf{W}_{f_s(x)}^l\}_l$  is a set of display scenario-specific parameters for  $f_s(x)$ . These parameters are then combined to form  $\theta(f_t(x), f_s(x))$ .

In this way, we avoid using samples that have both type  $f_t(x)$  and scenario  $f_s(x)$  to train a specific set of parameters as shown in



**Figure 4: Illustration of how auto-masking can allow a mini-batch of instances from different (conversion type, display scenario) domains to be computed simultaneously with the same operators but different domain-specific parameters. Back propagation is fast because no gradient needs to be computed for masked entries (i.e., setting to 0).**

Figure 3(a), and thus mitigate the data sparsity problem. Moreover, we only create  $N_t + N_s + 1$  sets of domain-specific parameters, but can generate  $N_t \times N_s$  combinations to serve all different (conversion type, display scenario) domains.

After we obtain the logit  $h$ , the predicted CVR is given by  $p_{cvr}(x) = \sigma(h)$ , where  $\sigma$  is the sigmoid function.

### 2.3 Auto-Masking

The above design seems fine. However, the model needs to choose specific parameters for each instance according to its conversion type and display scenario. In modern deep learning systems such as Tensorflow [1], instances are processed in mini-batches with the same operators. As a result, how to process a mini-batch of instances simultaneously with the *same* operators but allow each instance to use *different* domain-specific parameters is still a problem.

Previous works [4, 11, 21, 25] avoid this problem by splitting data into small per-domain datasets (Figure 3(a)). Accordingly, multiple input interfaces are created, one for each dataset. However, applying this approach to our problem will result in  $N_t \times N_s$  datasets, which incurs tremendous data processing, storage and maintenance cost.

We propose the following auto-masking strategy to solve this problem (Figure 4). Denote a mini-batch of  $N$  instances as  $X = [x_1, x_2, \dots, x_N]'$ , where  $'$  is the transpose operator. We create a mask vector  $\mathbf{m}_{t_i s_j}$  for each  $(t_i, s_j)$  domain, where  $t_i$  is the  $i$ th conversion type and  $s_j$  is the  $j$ th display scenario. The size of  $\mathbf{m}_{t_i s_j}$  equals the batch size  $N$ . In particular, we compute  $\mathbf{m}_{t_i s_j}$  as

$$\mathbf{m}_{t_i s_j}(X) = I[f_t(X) == t_i, f_s(X) == s_j],$$

where  $I(\cdot)$  is an indication function.

We show an example in Figure 4, where we have 4 instances,  $f_t(X) = [t_1, t_2, t_1, t_2]'$  and  $f_s(X) = [s_1, s_1, s_1, s_2]'$ . We then have

$$\begin{aligned} \mathbf{m}_{t_1 s_1}(X) &= [1, 0, 1, 0]', & \mathbf{m}_{t_1 s_2}(X) &= [0, 0, 0, 0]', \\ \mathbf{m}_{t_2 s_1}(X) &= [0, 1, 0, 0]', & \mathbf{m}_{t_2 s_2}(X) &= [0, 0, 0, 1]'. \end{aligned}$$

The mini-batch of instances go through all the CVR towers. Denote the output (i.e., mini-batch prediction) of the CVR tower for  $(t_i, s_j)$  domain as  $\mathbf{p}_{t_i s_j}(X)$ . Clearly,  $X$  may contain instances that

do not match  $(t_i, s_j)$  and these results are incorrect. For example, the mini-batch CVR prediction of tower  $(t_1, s_1)$  in Figure 4 is

$$\mathbf{p}_{t_1 s_1}(X) = [p_{t_1 s_1}(x_1, t_1, s_1), p_{t_1 s_1}(x_2, t_2, s_1), p_{t_1 s_1}(x_3, t_1, s_1), p_{t_1 s_1}(x_4, t_2, s_2)]',$$

where gray color indicates incorrect prediction because of parameter mismatch. We obtain the correct mini-batch CVR prediction  $\mathbf{p}_{\text{cvr}}(X)$  by combining the masked outputs of all different towers

$$\begin{aligned} \mathbf{p}_{\text{cvr}}(X) &= \sum_i \sum_j \mathbf{m}_{t_i s_j}(X) \odot \mathbf{p}_{t_i s_j}(X) \\ &= [p_{t_1 s_1}(x_1, t_1, s_1), 0, p_{t_1 s_1}(x_3, t_1, s_1), 0]' + [0, 0, 0, 0]' + \\ &\quad [0, p_{t_2 s_1}(x_2, t_2, s_1), 0, 0]' + [0, 0, 0, p_{t_2 s_2}(x_4, t_2, s_2)]' \\ &= [p_{t_1 s_1}(x_1, t_1, s_1), p_{t_2 s_1}(x_2, t_2, s_1), p_{t_1 s_1}(x_3, t_1, s_1), p_{t_2 s_2}(x_4, t_2, s_2)]', \end{aligned}$$

where  $\odot$  is element-wise product.

By using the auto-masking strategy, we allow a mini-batch of instances from different (conversion type, display scenario) domains to be computed simultaneously with the same operators but adaptively with different domain-specific parameters.

## 2.4 Training with Dynamically Weighted Loss

The training of MMN is much faster than that of multi-task learning models such as MMoE [17], because each entry of  $\mathbf{p}_{\text{cvr}}(X)$  only impacts the parameters of one tower rather than all the towers. No gradient needs to be computed during gradient back propagation for masked entries (i.e., setting to 0).

The training of the model may use the classical loss of ESMM [18], which combines a CTR loss and a CTCVR loss as

$$\text{loss} = \text{loss}_{\text{ctr}} + \alpha \text{loss}_{\text{ctcvr}},$$

where  $\alpha$  is a balancing parameter and

$$\begin{aligned} \text{loss}_{\text{ctr}} &= \frac{1}{N} \sum_{n=1}^N l_{\text{ctr}}(x_n), \quad \text{loss}_{\text{ctcvr}} = \frac{1}{N} \sum_{n=1}^N l_{\text{ctcvr}}(x_n), \\ l_{\text{ctr}}(x_n) &= -\{y_n \log[p_{\text{ctr}}(x_n)] + (1 - y_n) \log[1 - p_{\text{ctr}}(x_n)]\}, \\ l_{\text{ctcvr}}(x_n) &= -\{y_n z_n \log[p_{\text{ctr}}(x_n) p_{\text{cvr}}(x_n)] \\ &\quad + (1 - y_n z_n) \log[1 - p_{\text{ctr}}(x_n) p_{\text{cvr}}(x_n)]\}. \end{aligned} \quad (2)$$

$N$  is the batch size,  $x_n$  is the  $n$ th instance,  $y_n$  is the click label,  $z_n$  is the conversion label, and  $l_{\text{ctr}}, l_{\text{ctcvr}}$  are cross-entropy losses.

Further examination on the loss reveals that it is not quite suitable for MMN. Denote the number of CVR domains as  $C$ , the  $c$ th domain as  $\mathcal{D}_c$  and the number of instances that fall in the  $c$ th domain in a mini-batch as  $N_c$ . We further denote the expectation of the loss on each instance as  $\bar{a}$ , i.e.,  $E[l_{\text{ctcvr}}(x_n)] = \bar{a}$ . We then have

$$\begin{aligned} E(\text{loss}_{\text{ctcvr}}) &= \frac{1}{N} \sum_{n=1}^N E[l_{\text{ctcvr}}(x_n)] = \frac{1}{N} \left[ \sum_{c=1}^C \sum_{n=1}^{N_c} E[l_{\text{ctcvr}}(x_n \in \mathcal{D}_c)] \right] \\ &= \sum_{c=1}^C \left[ \frac{1}{N} \sum_{n=1}^{N_c} E[l_{\text{ctcvr}}(x_n \in \mathcal{D}_c)] \right] = \sum_{c=1}^C \left( \frac{N_c}{N} \bar{a} \right). \end{aligned}$$

That is, the average loss applied on each domain is actually  $\frac{N_c}{N} \bar{a} \leq \bar{a}$  and varies from one domain to another. For each domain, we actually do not average over the instances corresponding to it but use an incorrect normalization value  $N$ .

Therefore, we propose the dynamically weighted loss as

$$\text{loss}_{\text{MMN}} = \text{loss}_{\text{ctr}} + \alpha \text{loss}'_{\text{ctcvr}},$$

$$\begin{aligned} \text{loss}'_{\text{ctcvr}} &= \sum_{c=1}^C \left[ I(N_c > 0) \frac{1}{N_c} \sum_{n=1}^{N_c} l_{\text{ctcvr}}(x_n \in \mathcal{D}_c) \right] \\ &= \frac{1}{N} \sum_{c=1}^C \left[ I(N_c > 0) \frac{N}{N_c} \sum_{n=1}^{N_c} l_{\text{ctcvr}}(x_n \in \mathcal{D}_c) \right] \\ &= \frac{1}{N} \sum_{n=1}^N \text{wgt}(x_n) l_{\text{ctcvr}}(x_n), \end{aligned} \quad (3)$$

where  $\text{wgt}(x_n) = \frac{N}{N_c}$  if  $x_n \in \mathcal{D}_c$ .

Note that the weight  $\text{wgt}(x_n)$  is not pre-computed before training, but *dynamically computed within each mini-batch*. This is why we call the loss as dynamically weighted loss. In other words, even if two instances belong to the same domain, their weights could be different if they appear in different mini-batches because the number  $N_c$  of samples in that domain could vary in different mini-batches.

## 2.5 Fast Online Prediction

As the online prediction is on demand, there is no need to let an online request go through all the CVR towers and use auto-masking to get the correct prediction. Given an online request  $x$  with the corresponding conversion type  $f_t(x)$  and display scenario  $f_s(x)$ , we simply let  $x$  go through the tower for  $(f_t(x), f_s(x))$  to calculate a single predicted CVR, i.e., until a *colored node* in Figure 2(a). This makes the online prediction of MMN fast.

## 3 EXPERIMENTS

### 3.1 Datasets

The statistics of the datasets are listed in Table 1.

(1) **News feed conversion log dataset**. This dataset contains a random sample of 14 days of ad impression, click and conversion logs from an industrial news feed advertising system for UC Toutiao. It contains  $N_t = 19$  conversion types and  $N_s = 8$  display scenarios.

(2) **Criteo conversion log dataset**<sup>2</sup>. This dataset contains a sample of 90 days of Criteo live traffic data [26]. There are  $N_t = 21$  conversion types and  $N_s = 17$  display scenarios.

All the features are hashed. Each training set is further split into an initial training set (70%) and a validation set (30%) to find the optimal hyperparameters.

### 3.2 Methods in Comparison

We compare three categories of methods.

#### A. Single-task methods

- (1) **DNN**. Deep Neural Network [8]. It contains an embedding layer, several fully connected layers and an output layer.
- (2) **ESMM**. Entire Space Multi-task Model [18]. It models both CVR and CTR prediction tasks.

#### B. Multi-task methods

- (3) **MMoE**. Multi-gate Mixture-of-Experts model [17]. It substitutes the shared bottom network with an MoE layer. Each task has a separate gating network.
- (4) **PLE**. Progressive Layered Extraction model [27]. It separates shared components and task-specific components explicitly and adopts a progressive routing mechanism.

<sup>2</sup><https://ailab.criteo.com/criteo-sponsored-search-conversion-log-dataset/>

Table 1: Statistics of experimental datasets.

Dataset	# Fields	# Features	# Train insts.	# Test insts.	# Conv. types (CVR range)	# Display scens. (CVR range)
News Feed	58	61,601,087	153,300,184	5,179,132	19 (0.05% ~ 27.6%)	8 (1.6% ~ 4.1%)
Criteo	18	11,772,096	11,196,944	4,798,690	21 (6.3% ~ 18.9%)	17 (0.9% ~ 14.9%)

### C. Multi-domain methods

- (5) **MT-FwFM**. Multi-Task Field-weighted Factorization Machine [22]. As it models type-specific parameters for multi-type CVR prediction, we classify it as a multi-domain method.
- (6) **STAR**. Star Topology Adaptive Recommender [25]. It is proposed for CTR prediction over multiple business domains such as “Banner” and “Guess What You Like”.
- (7) **MMN**. Masked Multi-domain Network in this paper.

**Parameter Settings:** We set the dimension of the embedding vectors for features as 10. For neural network-based models, we set the number of fully connected layers in a CTR / CVR tower as  $L = 4$  with units {512, 256, 128, 128}. All the methods are implemented in Tensorflow [1] and optimized by the Adagrad algorithm [10]. We run each method 3 times and report the average results.

### 3.3 Evaluation Metrics

**Accuracy:** The Area Under the ROC Curve (AUC) is a widely used metric for CVR prediction [22]. The larger the better. We calculate fine-grained metrics as per conversion type AUC and per display scenario AUC. In addition, we also calculate the average AUC over all the conversion types and display scenarios.

**Scalability:** We use the number of sets of domain-specific parameters to evaluate the scalability.

**Convenience:** We use the number of separate datasets needed to train the model to evaluate the convenience.

### 3.4 Offline Performance

**3.4.1 Accuracy.** Table 2 and Table 3 list the test AUCs. As multi-task methods are not quite suitable for our problem setting, MMoE and PLE perform even worse than single-task methods such as DNN and ESMM on the Criteo dataset. On the News Feed dataset, their performance is comparable. Multi-domain method STAR outperforms single-task methods such as DNN and ESMM. MMN achieves the highest AUC in most cases. It is because MMN has the ability to tackle both conversion type and display scenario simultaneously by design. In contrast, other multi-task / multi-domain methods can either optimize on the flattened type  $\times$  scenario space with significant data sparsity issue, or optimize only one dimension because of the out-of-memory issue.

**3.4.2 Scalability.** We use the number of sets of domain-specific parameters to evaluate the scalability. Single-task methods do not model domain-specific parameters and do not have the scalability issue. MMN models  $N_t + N_s + 1$  sets while other multi-task / multi-domain methods model  $N_t \times N_s$  sets of domain-specific parameters. Compared with these methods, MMN reduces the number of sets of domain-specific parameters from 357 to 39 (i.e., 89.1% reduction) in the Criteo dataset and from 152 to 28 (i.e., 81.6% reduction) in the News Feed dataset.

**3.4.3 Convenience.** We use the number of separate datasets needed to train the model to evaluate the convenience. Single-task / multi-task methods use only one dataset and they are convenient. With the help of auto-masking, MMN also uses only one dataset. However, other multi-domain methods need to create a separate dataset for each domain and results in  $N_t \times N_s$  sub datasets. Compared with these methods, MMN reduces the number of separate datasets from 357 to 1 (i.e., 99.7% reduction) in the Criteo dataset and from 152 to 1 (i.e., 99.3% reduction) in the News Feed dataset.

**3.4.4 Ablation Study: effect of type-specific and scenario-specific parameters.** Figure 5 plots the AUC difference between MMN and MMN (common params.). In MMN (common params.), we only use shared common parameters  $\{\mathbf{W}_b^l\}_l$  for all the domains. In other words, no type-specific and scenario-specific parameters are learned. It is observed that MMN outperforms MMN (common params.) in most cases. These results show that different conversion types and display scenarios do have their own properties and using the same parameter set cannot well capture individual properties.

**3.4.5 Ablation Study: effect of dynamically weighted loss.** Figure 6 plots the AUC difference between MMN and MMN (no dynamic weight). In the latter, we use the loss in Eq.(2) for model training. It is observed that AUCs of MMN improve a lot compared with MMN (no dynamic weight) in most cases. These results validate the effectiveness of the dynamically weighted loss.

### 3.5 Online Deployment

We deployed MMN in an industrial news feed advertising system for UC Toutiao which serves hundreds of millions of traffic every day. MMN is trained using a distributed CPU cluster which contains 8 servers each with 96 CPU core units and 512 GB RAM. By design, MMN can satisfy the scalability and convenience requirements. In practice, we do not encounter the out-of-memory issue when training MMN but when training STAR on the type  $\times$  scenario space. Moreover, MMN needs only one dataset to train the model, and it significantly reduces the data processing overhead. We conducted online experiments in an A/B test framework over 14 days in August 2022, where the base serving model is ESMM. By design, the online prediction time of MMN is short. In practice, the prediction time of MMN is 2.3 ms, which is only slightly longer than that of ESMM, which is 2.2 ms. MMN leads to clear online CVR increases in all the 19 conversion types and the 8 display scenarios, where the increases range from 3.2% to 12.6%. Moreover, the CPM (revenue) is increased by 5.1%. After the A/B test, MMN provides real-time CVR prediction for all ad traffic in our system.

## 4 RELATED WORK

**CVR prediction.** The task of CVR prediction [6, 13, 15, 23] in online advertising is to estimate the probability of a user makes a conversion event on a specific ad after a click event. [13] estimates CVR based on past performance observations along data

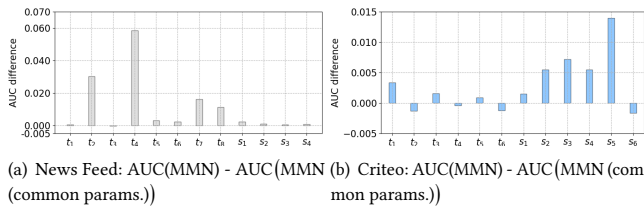


**Table 2: Test AUCs on the *News Feed* ad dataset. The best result is in bold font. Randomly selected conversion types / display scenarios that span the training CVR spectrum are shown. (type / scen.): a method optimizes only the type / scenario dimension because of the out-of-memory issue. A small improvement in AUC (e.g., 0.0020) can lead to a significant increase in online CVR.**

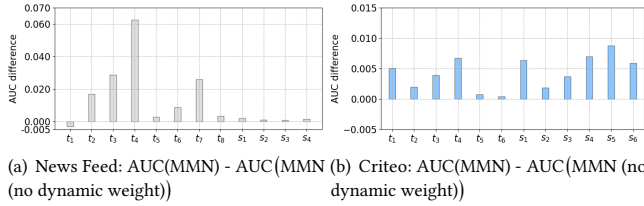
		Conversion types								Display scenarios				Avg.
	Type / Scen. code	50	44	65	18	1	11	5	66	3126	8082	5739	4661	
Method	CVR	27.6%	14.9%	3.1%	2.8%	1.3%	0.5%	0.3%	0.05%	4.1%	3.7%	2.1%	1.6%	
Single-task	DNN	0.8584	0.7247	0.6424	0.7898	0.8402	0.7941	0.7815	0.6825	0.9688	0.8112	0.9346	0.9698	0.8165
	ESMM	0.8576	0.7253	0.6519	0.7906	0.8404	0.7927	0.7852	0.6829	0.9687	0.8114	0.9343	0.9697	0.8176
Multi-task	MMoE (type)	0.8574	0.7157	0.6565	0.7953	0.8406	0.7909	0.7861	0.6796	0.9686	0.8110	0.9340	0.9698	0.8171
	MMoE (scen.)	0.8574	0.6994	0.6449	0.7921	0.8385	0.7908	0.7824	0.6792	0.9683	0.8110	0.9330	0.9693	0.8139
	PLE (type)	0.8578	0.7258	0.6499	0.7948	0.8411	0.7916	0.7865	0.6832	0.9685	0.8116	0.9346	0.9698	0.8179
	PLE (scen.)	0.8577	0.7231	0.6525	0.7931	0.8402	0.7931	0.7852	0.6976	0.9687	0.8118	0.9342	0.9701	0.8189
Multi-domain	MT-FwFM	0.8508	0.6670	0.6273	0.7764	0.8272	0.7648	0.7585	0.5835	0.9676	0.8004	0.9288	0.9663	0.7932
	STAR (type)	0.8616	0.7554	0.6687	0.7832	0.8419	0.7952	0.7865	0.7018	0.9691	0.8125	0.9353	0.9703	0.8235
	STAR (scen.)	0.8583	0.7550	0.6642	0.7924	0.8417	0.7946	<b>0.7868</b>	0.6677	0.9691	0.8148	0.9380	0.9703	0.8211
	MMN	<b>0.8626</b>	<b>0.7564</b>	<b>0.6842</b>	<b>0.7962</b>	<b>0.8427</b>	<b>0.8063</b>	0.7865	<b>0.7219</b>	<b>0.9710</b>	<b>0.8168</b>	<b>0.9411</b>	<b>0.9732</b>	<b>0.8299</b>

**Table 3: Test AUCs on the *Criteo* ad dataset. The best result is in bold font.**

		Conversion types						Display scenarios						Avg.
	Type / Scen. code	7647	1208	4445	6198	7652	1584	5264	8441	1132	9036	1438	8972	
Method	CVR	18.9%	13.2%	11.4%	9.6%	8.0%	6.3%	14.9%	13.5%	12.0%	10.2%	6.4%	0.9%	
Single-task	DNN	0.7606	0.7556	0.7638	0.7762	0.8220	0.7376	0.7310	0.6383	<b>0.6191</b>	0.7402	0.6579	0.7708	0.7311
	ESMM	0.7593	0.7538	0.7645	0.7707	0.8213	0.7405	0.7310	0.6310	0.6156	0.7365	0.6547	0.7792	0.7298
Multi-task	MMoE (type)	0.7129	0.7484	0.7556	0.7681	0.8176	0.5444	0.7028	0.6212	0.6089	0.7030	0.5727	0.7359	0.6910
	MMoE (scen.)	0.7659	0.7463	0.7460	0.7686	0.8168	0.7238	0.7239	0.5469	0.5032	0.7162	0.5033	0.7596	0.6934
	PLE (type)	0.7468	0.7512	0.7593	0.7729	0.8212	0.7323	0.7267	0.6297	0.6123	0.7371	0.6511	0.7683	0.7257
	PLE (scen.)	0.7565	0.7526	0.7606	0.7731	0.8224	0.7372	0.7255	0.6321	0.6105	0.7381	0.6432	0.7752	0.7273
Multi-domain	MT-FwFM	0.7491	0.7345	0.7294	0.7528	0.8034	0.7112	0.6831	0.6078	0.5858	0.6788	0.5953	0.7239	0.6963
	STAR (type)	0.7692	0.7585	0.7663	0.7776	0.8237	0.7445	0.7340	0.6547	0.6176	0.7356	0.6648	0.7844	0.7359
	STAR (scen.)	0.7715	0.7565	0.7644	0.7680	0.8226	0.7458	0.7352	<b>0.6611</b>	0.6155	0.7373	0.6614	0.7878	0.7356
	MMN	<b>0.7741</b>	<b>0.7586</b>	<b>0.7689</b>	<b>0.7834</b>	<b>0.8251</b>	<b>0.7473</b>	<b>0.7370</b>	0.6569	0.6158	<b>0.7413</b>	<b>0.6747</b>	<b>0.7895</b>	<b>0.7394</b>



**Figure 5: Effect of domain-specific parameters.**



**Figure 6: Effect of dynamically weighted loss.**

hierarchies. [7] proposes an LR model for CVR prediction and [2] proposes a log-linear model. [18] proposes the ESMM model to jointly consider CVR and CTR prediction tasks and exploit data in the entire sample space. ESM<sup>2</sup> [29], GMCM [3] and HM<sup>3</sup> [28] exploit additional purchase-related behaviors after click (e.g., favorite, add to cart and read reviews) for CVR prediction. But such behaviors are not always available in different advertising systems.

**Multi-task/multi-domain learning.** In multi-task learning [14, 19, 24], the domains are the same and the tasks are different. In multi-domain learning [9, 12, 30], the domains are different and the tasks are the same. Because of such differences, existing multi-task learning models such as Shared-Bottom, OMoe, MMoe, CGC and PLE [16, 17, 27, 31] cannot well address our problem. MT-FwFM [22], STAR [25] and ADIN [11] are multi-domain models that are more closely related to our work. However, they cannot address both different conversion types and different display scenarios. Moreover, they are not designed to satisfy the scalability and convenience requirements for our problem.

## 5 CONCLUSION

In this paper, we present the design and evaluation of masked multi-domain network (MMN) to address the multi-type and multi-scenario CVR prediction problem in a real-world advertising system. We design various strategies such as modeling domain-specific parameters, parameter sharing and composition, auto-masking and dynamically weighted loss in order that MMN can simultaneously satisfy the accuracy, scalability and convenience requirements as an industrial model. Experimental results show that MMN does achieve these requirements and it outperforms several state-of-the-art models for multi-type and multi-scenario CVR prediction.

## REFERENCES

- [1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. USENIX, 265–283.
- [2] Deepak Agarwal, Rahul Agrawal, Rajiv Khanna, and Nagaraj Kota. 2010. Estimating rates of rare events with multiple hierarchies through scalable log-linear models. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 213–222.
- [3] Wentian Bao, Hong Wen, Sha Li, Xiao-Yang Liu, Quan Lin, and Keping Yang. 2020. GMCM: Graph-based micro-behavior conversion model for post-click conversion rate estimation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2201–2210.
- [4] Jiangxia Cao, Shaoshuai Li, Bowen Yu, Xiaobo Guo, Tingwen Liu, and Bin Wang. 2023. Towards Universal Cross-Domain Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM)*. 78–86.
- [5] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.
- [6] Olivier Chapelle. 2014. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1097–1105.
- [7] Olivier Chapelle, Eren Manavoglu, and Romer Rosales. 2014. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 4 (2014), 1–34.
- [8] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishu Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ipsir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS)*. ACM, 7–10.
- [9] Mark Dredze, Alex Kulesza, and Koby Crammer. 2010. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning* 79, 1 (2010), 123–149.
- [10] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.
- [11] Yuchen Jiang, Qi Li, Han Zhu, Jinbei Yu, Jin Li, Ziru Xu, Huihui Dong, and Bo Zheng. 2022. Adaptive Domain Interest Network for Multi-domain Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*. 3212–3221.
- [12] Mahesh Joshi, Mark Dredze, William Cohen, and Carolyn Rose. 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.
- [13] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past performance data. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 768–776.
- [14] Ziru Liu, Jiejie Tian, Qingpeng Cai, Xiangyu Zhao, Jingtong Gao, Shuchang Liu, Dayou Chen, Tonghao He, Dong Zheng, Peng Jiang, et al. 2023. Multi-Task Recommendations with Reinforcement Learning. In *Proceedings of the ACM Web Conference 2023 (WWW)*. 1273–1282.
- [15] Quan Lu, Shengjun Pan, Liang Wang, Junwei Pan, Fengdan Wan, and Hongxia Yang. 2017. A practical framework of conversion rate prediction for online display advertising. In *Proceedings of the International Workshop on Data Mining for Online Advertising (ADKDD)*. 1–9.
- [16] Jiaqi Ma, Zhe Zhao, Jilin Chen, Ang Li, Lichan Hong, and Ed H Chi. 2019. SNR: Sub-network routing for flexible parameter sharing in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 216–223.
- [17] 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)*. 1930–1939.
- [18] 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 (SIGIR)*. ACM, 1137–1140.
- [19] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*. 3994–4003.
- [20] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*. 807–814.
- [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)*. ACM, 2669–2676.
- [22] Junwei Pan, Yizhi Mao, Alfonso Lobos Ruiz, Yu Sun, and Aaron Flores. 2019. Predicting different types of conversions with multi-task learning in online advertising. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 2689–2697.
- [23] Römer Rosales, Haibin Cheng, and Eren Manavoglu. 2012. Post-click conversion modeling and analysis for non-guaranteed delivery display advertising. In *Proceedings of the fifth ACM international conference on Web Search and Data Mining (WSDM)*. 293–302.
- [24] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).
- [25] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, et al. 2021. One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM)*. 4104–4113.
- [26] Marcelo Tallis and Pranjul Yadav. 2018. Reacting to Variations in Product Demand: An Application for Conversion Rate (CR) Prediction in Sponsored Search. *arXiv preprint arXiv:1806.08211* (2018).
- [27] 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 *Proceedings of the 14th ACM Conference on Recommender Systems (RecSys)*. 269–278.
- [28] Hong Wen, Jing Zhang, Fuyu Lv, Wentian Bao, Tianyi Wang, and Zulong Chen. 2021. Hierarchically modeling micro and macro behaviors via multi-task learning for conversion rate prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2187–2191.
- [29] 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 (SIGIR)*. 2377–2386.
- [30] Xuanhua Yang, Xiaoyu Peng, Penghui Wei, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. AdaSparse: Learning Adaptively Sparse Structures for Multi-Domain Click-Through Rate Prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*. 4635–4639.
- [31] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*. 43–51.
- [32] 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 (KDD)*. 2191–2200.