Leaving No One Behind: A Multi-Scenario Multi-Task Meta Learning Approach for Advertiser Modeling

Qianqian Zhang, Xinru Liao, Quan Liu*, Jian Xu, Bo Zheng

Alibaba Group, China

{qianfan.zqq,xinru.lxr,lq204691,xiyu.xj,bozheng}@alibaba-inc.com

ABSTRACT

Advertisers play an essential role in many e-commerce platforms like Taobao and Amazon. Fulfilling their marketing needs and supporting their business growth is critical to the long-term prosperity of platform economies. However, compared with extensive studies on user modeling such as click-through rate predictions, much less attention has been drawn to advertisers, especially in terms of understanding their diverse demands and performance. Different from user modeling, advertiser modeling generally involves many kinds of tasks (e.g. predictions of advertisers' expenditure, active-rate, or total impressions of promoted products). In addition, major e-commerce platforms often provide multiple marketing scenarios (e.g. Sponsored Search, Display Ads, Live Streaming Ads) while advertisers' behavior tend to be dispersed among many of them. This raises the necessity of multi-task and multi-scenario consideration in comprehensive advertiser modeling, which faces the following challenges: First, one model per scenario or per task simply doesn't scale; Second, it is particularly hard to model new or minor scenarios with limited data samples; Third, inter-scenario correlations are complicated, and may vary given different tasks.

To tackle these challenges, we propose a multi-scenario multi-task meta learning approach (M2M) which simultaneously predicts multiple tasks in multiple advertising scenarios. Specifically, we introduce a novel meta unit that incorporates rich scenario knowledge to learn explicit inter-scenario correlations and can easily scale to new scenarios. Furthermore, we present a meta attention module to capture diverse inter-scenario correlations given different tasks, and a meta tower module to enhance the capability of capturing the representation of scenario-specific features. Compelling results from both offline evaluation and online A/B tests demonstrate the superiority of M2M over state-of-the-art methods.

CCS CONCEPTS

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

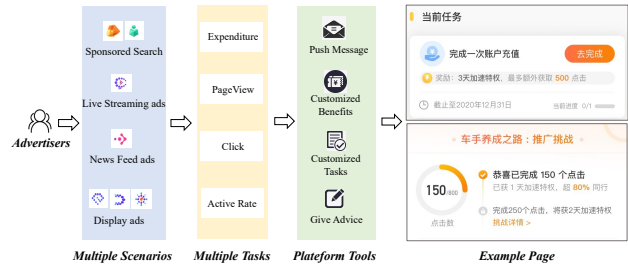*Corresponding author: Quan Liu, lq204691@alibaba-inc.com.

Figure 1: The interaction flow between advertisers and the ad platform: 1) Advertisers operate in multiple scenarios; 2) Our model routinely generates multi-task predictions of each advertiser in corresponding scenarios; 3) These predictions feed into various advertising engagement tools.

KEYWORDS

Advertiser Modeling, Multi-Task Learning, Meta Learning, Multi-Behavior Learning

1 INTRODUCTION

In the Internet era, large e-commerce platforms such as Taobao and Amazon are becoming primary places for users to find and purchase products. With many potential consumers on e-commerce platforms, an increasing proportion of sellers are packed into e-commerce platforms to advertise their products. Although advertisers play an important role in this ecosystem, much less attention has been paid to understanding advertisers either from academic or industrial communities. Existing studies mainly focus on the user side [2, 7, 17, 27, 28, 40, 41], while some [8][37] noticed the necessity of advertiser understanding for platforms' long-term development, but they focused on predicting one single task like churn rate. As different advertisers at different business cycles have various demands as well as advertising performance, for example, impressions or clicks of advertised products, ROI (i.e return on investment), expenditure constraints, active or churn rate, etc, it is insufficient to measure overall conditions of advertisers based on one single task.

Apart from multi-task demands, multi-scenario also needs to be considered in advertiser modeling. As illustrated in Figure 1, an advertiser may operate in multiple advertising scenarios such as sponsored search, news feed ads and even new emerging ones
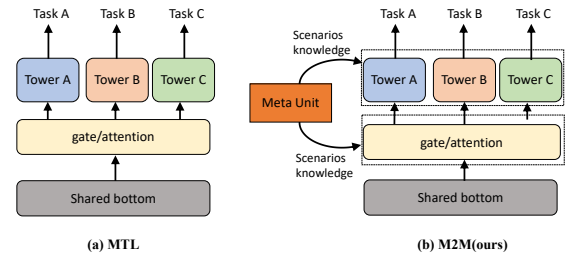
Table 1: The pearson correlations of different pair of scenarios given different tasks.

Pearson Correlations	Task.Expenditure	Task.Click
<Sponsored Search, News Feed Ads>	0.67	0.61
<Sponsored Search, Display Ads>	0.57	0.53
<News Feed Ads, Display Ads>	0.70	0.45

like live streaming ads (e.g. Taobao Live). A thorough modeling of advertisers require predicting advertisers' various tasks mentioned above in each scenario, without which we may end up in partial conclusions about advertisers. And this comprehensive understanding further enables advertising platforms to generate customized treatments like Rewarded Task and Push Message to better engage advertisers, which is quite essential for long-term development of platform economies.

Modelling advertisers in a multi-scenario multi task fashion turns out to be far from trivial. In particular, it faces challenges including (1) **Limited reference samples in new or minor scenarios**. The number of training samples can be quite different for different scenarios, with some minor scenarios only having a few thousand daily active advertisers. A conventional solution would train exclusive models for each scenario, while the data from small scenarios are too sparse to train a reliable model. This problem worsens when predicting in new scenarios. How to transfer information among different scenarios while maintaining each's own specific characteristics is not trivial. (2) **Complex inter-scenario relations with multiple tasks**. Inter-scenario correlations are complex and not easy to precisely capture. Given a task like prediction of clicks or expenditure of advertisers in different scenarios, the inter-scenario correlations can sometimes be positive, negative, or unrelated. Furthermore, these correlations may evolve and vary when modeling different tasks. As illustrated in Table 1, for the same task in different pair of marketing scenarios, the correlation coefficients are quite different. One common solution here is multiple task learning (MTL), which models the correlations of multiple tasks in one scenario, and learns simultaneously in one model and is proven to improve learning efficiency through information sharing between tasks [20][33][22][18][19]. However, these MTL-based methods cannot explicitly or effectively model multiple scenarios beyond multiple tasks. How to capture the complex inter-scenario correlations with multiple tasks to improve prediction performance of multiple scenarios is another key challenge.

To address these challenges, we propose M2M, a multi-scenario multi-task meta learning model, to predict multiple tasks in multiple scenarios all in a fully automatic manner. First, a backbone network is adopted to learn representations of advertiser-related features and task representations. Then, to learn the meta-knowledge of scenario-related information explicitly, we design a Meta Unit with customized scenario-related input, and its output is used to generate layer weights for later prediction networks. Furthermore, to combine multiple-scenario multiple-task in an end-to-end fashion effectively, we propose a Meta Learning Mechanism, which comprises a meta attention layer and a meta residual tower layer,

Figure 2: Different Solution: Figure(a) is the conventional deep model of multi-task learning which incorporates shared-bottom expert structures and gating network to learn expert representations, and feeds into a specific tower network. Figure(b) additionally uses meta unit of scenario knowledge to generate network weights for different scenarios.

respectively to capture diverse inter-scenario correlations and enhance scenario-specific feature representation capability. Finally, we utilize the joint Poisson loss for multiple tasks to optimize our model in an end-to-end manner. Compelling results from both offline evaluation and online A/B tests demonstrate the superiority of M2M over state-of-the-art methods, especially for long-tail scenarios.

The contributions of this paper are highlighted as follows:

- We study the important but under-explored problem of comprehensive advertisers modeling to engage advertisers and help them grow, which is essential to the prosperity of advertising platforms. To the best of our knowledge, we are the first attempt to explicitly use different scenario-related information and propose a unified framework for advertiser understanding.
- We propose M2M, a novel multi-scenario multi-task paradigm, to predict various tasks from various scenarios in an end-to-end fashion. The key idea is that we combine the advantage of meta learning and multi-task learning, which enhances the ability of capturing diverse inter-scenario correlations given different tasks. Moreover, this approach easily scales to new scenarios and is proven to significantly improve the performance in new or minor scenarios.
- Compelling results from both offline evaluation and online A/B tests demonstrate the superiority of M2M over state-of-the-art methods, especially for emerging and minor scenarios. And our model has been successfully deployed in various advertising-related applications at Taobao.

2 RELATED WORK

User Modeling. In recommendation systems and digital advertising space, there is extensive work in modeling user intents and satisfaction like CTR(click-through rate)[41][27][11][10] and CVR(post-click conversion rate), which is crucial for matching and ranking systems. However, there is much less work on modeling advertisers' various aspects like impressions, clicks, GMV, active-rate, etc. [37] presents a method that carefully selects a homogeneous group of advertisers and uses classification algorithms to predict advertisers'

churn. [8] proposes a novel two-stage network to model advertiser intent and satisfaction simultaneously to provide better services for advertisers, but scenarios related information was not considered. Compared with user intent, modeling advertisers' behavior and performance is actually much more complicated for several reasons. First, user intents are generally more explicit, represented by clicks and purchases, while advertiser intents are harder to identify due to larger variety. For example, some may aim at controlling costs while others thrive for more impressions and sales. Second, user-side usually has rich data and uniformed behavior, while advertisers' operations are more sparse and diverse.

Multi-task Learning. Multi-task learning[30][38] is a machine learning paradigm that aims to leverage useful information across multiple tasks to improve performance of each task, which has many applications in various areas such as natural language processing[32], computer vision[15][3] and speech recognition[4][29]. In early literature of multi-task learning, [5] proposes a kernel method and regularization model to learn multi-task simultaneously. In the deep learning era, various deep representation learning approaches have been proposed for multi-task learning problems. [34] proposes to share parameters of hidden layers which is a commonly used approach. And [22] proposes a cross-stitch unit that learns to combine task-specific layer for each task linearly. [20] proposes Multi-gate Mixture-of-Experts(MMoE) to model the relationship across different tasks by sharing experts at the bottom layer, while utilizing different gate networks for each task to obtain fusing information of experts. CGC[33] develops further on the basis of MMoE, separating task-common and task-specific parameters explicitly to avoid parameter conflict resulting from complex task correlations. [15] defines variance uncertainty in loss to optimize the weights in multi-task learning. Despite their respective success in capturing multiple tasks relationship, to the best of our knowledge, few of these works considers modeling scenarios information explicitly.

Meta Learning. Meta-learning[14][26][31][34][35], or learning to learn, is the science of systematically observing how different machine learning approaches perform in multiple tasks, and learning from this experience to seek rapid and accurate model adaptation to unseen tasks, even with few training examples. It has been widely used in many applications like highly automated AI, few-shot learning, natural language processing[32] and image processing[6][23]. Meta-learning can help improve transfer learning and multi-task learning by using a meta-objective. MAML[6] and Reptile[24] propose an optimization-based and model-agnostic algorithm that uses meta-gradient updates to train the parameters explicitly so that a small number of gradient steps produces good generalization performance on new tasks. [23] introduces a novel MetaNet structure, which consists of two main learning components: the base learner performs in the input task space whereas the meta learner operates in a task-agnostic metaspace. [9]explores a more efficient method that uses a smaller network named hyper network to generate the structure of weights for a more extensive network, in order to search within the much smaller weight space.[1] constructs an efficient feed-forward one-shot learner, which demonstrates that a deep neural network can learn at the "meta-level" of predicting filter parameters for another main network. ST-MetaNet [25] employs sequence-to-sequence architecture, leveraging the meta-knowledge extracted from geography attributes to generate parameter weights

Table 2: Feature types and descriptions

Feature Type	Description
Scenario Attributes	Scenario Type, Statistic Scenario Characteristics, etc.
Advertiser Profile	Business Scale, Main Category, Star Level, etc.
Multi-type Behavior	Login, Campaign Modification, Bidding, Budget, etc.
Multi-type Performance	Expenditure, GMV, ROI, impressions etc.

for a graph attention network that captures diverse spatial correlation and a recurrent network that considers diverse temporal correlation. Unlike above works, we explicitly consider multi-level scenario information in one model, and combine the advantage of multi-task learning and meta learning as shown in Figure 2.

3 PRELIMINARIES

3.1 Problem Settings

Advertisers accumulate multiple types of behavior and performance in different advertising scenarios. To fully represent advertisers, we also establish advertiser profile and summarize scenario knowledge. More details about our used features are shown in Table 2.

DEFINITION 1. **Scenario Attributes**. *Let* $S = \{s_1, s_2, \ldots, s_l\}$ *denote the set of scenarios related information which is advertisers' marketing scenario like sponsored search, display ads, etc.*

DEFINITION 2. **Advertiser Profile**. *Let* $A = \{a_1, a_2, \ldots, a_m\}$ *denote the set of advertiser profile information.*

DEFINITION 3. **Multi-type Behavior Sequences**. *Let* $X_b = \{X_b^t\}_{t=1}^T$ *denote time-dependent multi-type behavior information in a time period T, where* $X_b^t = \{x_{b1}^t, x_{b2}^t, \ldots, x_{bm}^t\}$ *denote dense or sparse features of advertisers at time step t.*

DEFINITION 4. **Multi-type Performance Sequences**. *Let* $X_p = \{X_p^t\}_{t=1}^T$ *denote time-dependent multi-type performance information in a time period T, where* $X_p^t = \{x_{p1}^t, x_{p2}^t, \ldots, x_{pm}^t\}$ *denote dense or sparse features of advertisers at time step t.*
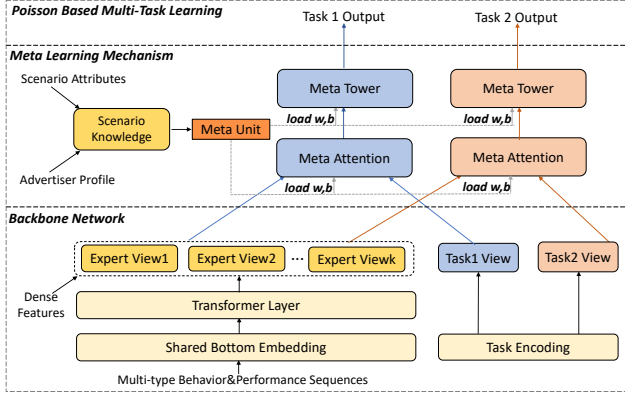
Multi-Scenario Multi-Task Prediction. Given scenario attributes S, advertiser profile A, multi-type behavior sequences X_b, multi-type performance sequences X_p, we need to train a unified model, to predict multiple tasks (e.g. expenditure, active-rate, clicks) in multiple scenarios(e.g. Sponsored Search, Display Ads, Star Shop) for the next τ days, which can be formulated as:

$$\hat{Y}_{ij}^{t+\tau}\,|_{scenarios_i, task_j} = \Gamma(S, A, X_b, X_p) \tag{1}$$

where d is the function we aim to learn, note that the proposed method can be easily extended to any time interval prediction.

4 THE PROPOSED M2M MODEL

In this section, we introduce our Multi-Scenario Multi-Task advertiser Modelling(M2M) approach which consists of three main stages. The first stage, *Backbone Network*, aims to obtain feature and task representations. The second stage *Meta Learning Mechanism* comprises a meta attention module and meta tower module, to capture diverse inter-scenario correlations and enhance scenario-specific feature representation capability respectively. The third

Figure 3: Overview of M2M Framework. The backbone network aims to obtain feature and task representations. The meta learning mechanism uses rich scenario knowledge to generate meta network weights of meta attention module and meta tower module. And the poisson-based multi-task learning uses poisson regression to generate predictions.

stage *Poisson Based Multi-Task Learning* is introduced to predict multiple tasks from multiple scenarios by Poisson loss. The overall architecture is depicted in Figure 3.

4.1 Backbone Network

4.1.1 Shared Bottom Embedding. Since these input sequence features $\{X_b, X_p\}$ include continuous features like the page view, click, or expenditure of different marketing scenarios, the first step is discretization. After discretization, the inputs are high dimensional binary vectors. We use embedding layer to transform them into low-dimensional dense representations. Finally, we obtain the embedding with fixed-size low-dimensional vectors of each time sequence, and then we can use the transformer layer to embed these performance and behavior sequence features separately. Note that we also use positional embedding for each time sequence to capture the order or time information in historical performance or behavior sequence. Here we concatenate the trainable position embedding into each time sequence embedding feature.

4.1.2 Transformer Layer. After embedding these features into low dimensional representations, we use transformer layer to learn the deeper representation of each time sequence by capturing the relations with other time sequence features, which are more computationally efficient and expressively more powerful compared with RNN and LSTM[13].
Self-attention Layer. We use the scaled dot-product attention which is defined in [36].

$$Attention(Q, K, V) = softmax(\frac{QK^\top}{\sqrt{d}}V), \qquad (2)$$

where Q represents the query, K the key, V the value of input embedding, and d the scale factor is the dimension of the key. The three matrices are the equal-dimensional linear projects of the input time sequence embedding features. Following [36] we use

multi-head self-attention:

$$MH(X) = concat(head_1, head_2, ..., head_h)W^H, \qquad (3)$$

$$head_i = Attention(XW_i^Q, XW_i^K, XW_i^V), \qquad (4)$$

where h is the number of heads, X the processed input embedding features of performance or behavior, W_i^Q, W_i^K and W_i^V are the project matrices and trainable parameters of $head_i$ respectively. Note that since we have the following expert view layer explained in 4.1.3, we doesn't incorporate additional none-linear units in the transformer layer. After we use transformer layer to learn to representation of behavior and performance sequence feature respectively, we concatenate then into a fixed-length vector.

$$F = concat(MH(X_b), MH(X_p)) \qquad (5)$$

4.1.3 Expert View Representation. After the transformer layer, we obtain the representation of the input feature F, which we can also concatenate other dense feature of advertisers. And following MMoE network structure, we build mixture-of-experts layers which obtain different shared representations.

$$E_i = f_{MLP}(F), \forall i \in 1, 2, .., k \qquad (6)$$

where E_i the ith expert output, k the number of expert views.

4.1.4 Task View Representation. Besides embedding features, inspired by MRAN[39], we also embed all the tasks in the same space, which act as the "anchor points" of the tasks and introduce prior knowledge of tasks and influence the weight of feature information. Note that there is no specific label information in the test set. Therefore, the task representation we extract is a global one rather than local. We first use the lookup table to transform one-hot task information into low dimensional dense representations; Then, a feed-forward layer is adopted to reshape the dimension. The feed-forward sub-layer is composed of a nonlinear layer with LeakyReLU[21] activation. Formally, we have the following equation :

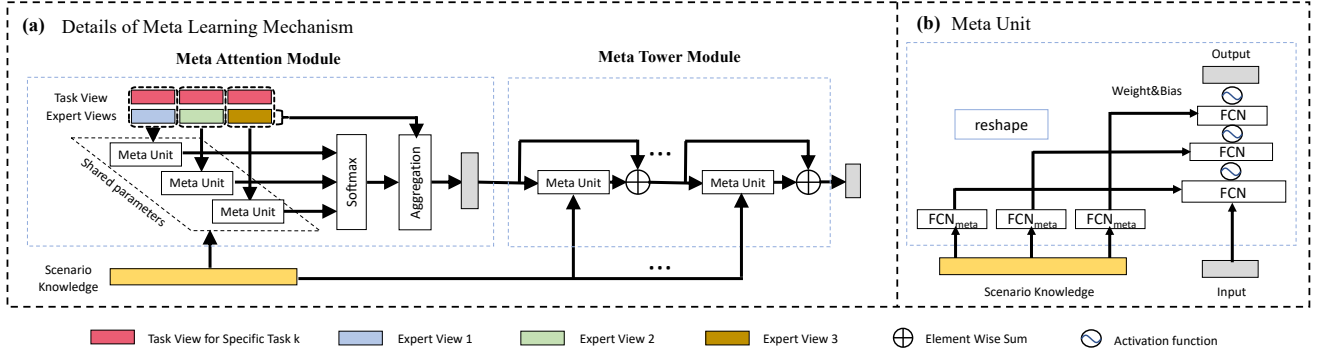$$T_t = f_{MLP}(Embedding(t)), \forall t \in 1, 2, .., m \qquad (7)$$

where T_t denotes the anchor embedding of task t, m the number of tasks.

4.1.5 Scenario Knowledge Representation. Scenario knowledge representation is the knowledge representation of the advertiser's advertising scenarios. Here we not only use the scenario attributes that the advertiser marketing, but also the advertiser profile information to better learn the multiple tasks. We incorporate a feed-forward layer to learn to representation of scenario knowledge.

$$\tilde{S} = f_{MLP}(S, A) \qquad (8)$$

4.2 Meta Learning Mechanism

To better characterize the scenario-specific representation from different sequential features, we propose a meta learning mechanism that consists of two components: a meta attention module and a meta residual tower module. These two modules are hierarchically organized. The meta attention module is positioned lower to capture diverse inter-scenario correlations, and a meta residual tower module is placed upper to enhance the capability in capturing the scenario-specific feature representation, as shown in Figure 4.

Figure 4: The Meta Learning Mechanism of our proposed M2M Framework. The figure shows the pipeline of one task which contains meta attention module and meta tower module. The meta attention module combines corresponding task view and expert views using network weights generated by specific scenario knowledge to get weighted representations of expert views. The meta tower module here uses residual tower network whose weights are also generated by corresponding scenarios knowledge.

4.2.1 Meta Unit. In our meta learning mechanism, we propose a meta unit to model different scenario-related information explicitly. As shown in Figure 4, to better capture inter-scenario correlations, we use scenarios knowledge $\tilde{S}$ as the input of meta unit. The meta unit transforms the scenario knowledge into the dynamic weights and bias parameters for meta attention learning and meta tower learning, which can be formulated as :

$$h^{(0)} = h_{input}, \tag{9}$$

$$h^{(i)} = \sigma(W^{(i-1)}h^{i-1} + b^{(i-1)}), \forall i \in 1, 2, .., K \tag{10}$$

where h_{input} is the input vector with dimension d, σ is a non-linear activation function. K is the total layer number of meta unit. $W^{(i)} \in \mathcal{R}^{d \times d}$ and $b^{(i)} \in \mathcal{R}^{d}$ are projection parameters. In particular, these projection parameters are generated by the meta unit from the scenario knowledge as shown in Figure 4. Formally:

$$W^{(i-1)} = Reshape(V_w\tilde{S} + v_w), \tag{11}$$

$$b^{(i-1)} = Reshape(V_b\tilde{S} + v_b), \tag{12}$$

Where $\tilde{S}$ represents scenario knowledge, after a fully connected network, a *Reshape* function is adopted to obtain weight matrix $W^{(i)}$ and bias vector $b^{(i)}$.

$$h_{output} = h^K = Meta(h_{input}), \tag{13}$$

Finally, the output of meta unit is obtained as h_{output}, we formally define the meta unit process as a *Meta* function.

Different from conventional works that add scenario related information as input features to models, our meta unit explicitly models inter-scenario correlations by incorporating scenario knowledge and producing dynamic parameter weights for each specific scenario. Through obtaining meta-representation of specific scenarios, the meta unit is proven to significantly improve the expressive power in new or minor scenarios.

4.2.2 Meta Attention Module. The meta attention module at the bottom tries to learn the contributing weights of feature views derived from the backbone network. Intuitively, each task may rely on different parts of the shared feature views. However, it is inappropriate to directly calculate the attention score ignoring the scenario factors. For instance, the amount of training data may

differ for different scenarios, and the data distribution of different scenarios may be dynamic for a specific task. While traditional alignment attention module can model the correlation of tasks and feature views, scenarios-specific diversity tends to be overlooked. Hence, we propose augmenting our M2M framework to capture the scenario signals while calculating the attention score, which can help the attention module learn how to produce dynamic attention weight in different scenarios. In specific, to model different patterns of attention score given the scenarios related information, different from traditional concatenation-based attention, we employ the above section meta unit, which is formulated as :

$$a_{t_i} = v^T Meta_t([E_i \parallel T_t]), \tag{14}$$

$$\alpha_{t_i} = \frac{exp(a_{t_i})}{\sum_{j=1}^{M} exp(a_{t_j})}, \tag{15}$$

where E_i is the sub-view feature vector of dimension d_1, T_t is the embedding of task anchor embedding of dimension d_2, σ is a non-linear activation function. The hidden vector $v \in \mathcal{R}^{(d_1+d_2)}$ projects the hidden vector into a scalar weight for output.

The final representation R_t for corresponding task t is the summation of the multiple sub-views weighted by their attention weights:

$$R_t = \sum_{i=1}^{k} \alpha_{t_i} E_i \tag{16}$$

Since we extract scenario knowledge and use such information to generate the weights and bias of the meta attention module, this module not only can model tasks and features' relevance, but also has the ability to capture the complex inter-scenario correlations.

4.2.3 Meta Tower Module. After obtaining the aggregated latent representations from the meta attention module, our M2M adopts a meta residual tower module[12] which also contains meta unit to distinguish different scenarios. Since feature patterns may vary from scenario to scenario for a specific task, a simple shared feedforward network is not sufficient to capture diverse scenarios information. To model such diversity, we introduce meta residual tower module as follows :

$$L_t^{(0)} = R_t, \tag{17}$$

Table 3: Dataset Details.

Dataset	Sce.A	Sce.B	Sce.C	Sce.D	Sce.E
Description	Sponsor Search	Display	Star Shop	News Feed	Brand Search
Train Data	25.28M	0.51M	0.47M	7.29M	0.84M
Valid Data	5.36M	0.12M	0.11M	1.50M	0.16M
Test Data	5.21M	0.11M	0.10M	1.52M	0.15M

$$L_t^{(j)} = \sigma(Meta^{(j-1)}(L_t^{(j-1)}) + L_t^{(j-1)}), \forall j \in 1, 2, .., L \qquad (18)$$

where R_t is the attention output of sub-views from different views, σ is a non-linear activation function. L is the number of residual layer.

4.3 Algorithm of Optimization

To train M2M jointly in an end-to-end manner, we adopt a common formulation of joint loss by calculating the weighted sum of the losses for each individual task with input X and task-specific labels $Y^t \in Y, t = 1, 2, ..., m$, which is defined as :

$$\mathcal{L}_{train}(S, A, X_b, X_p, Y) = \sum_{t=1}^{m} \lambda_t \mathcal{L}^t(S, A, X_b, X_p, Y^t) + \\ \alpha(\|W_1\|_F^2 + \|W_2\|_F^2) \qquad (19)$$

where the first term is a loss function, which is a linear combination of task-specific losses $\mathcal{L}^t$ with task weightings λ_t. The latter term is to penalize the complexity of W_1 and W_2 which denotes parameters from meta unit and other network weights respectively, and α is the regularization weight.

Poisson loss [8] is a widely used loss function for discrete count data, for each task t, we optimize the following objective function:

$$\mathcal{L}^t(S, A, X_b, X_p, Y^t) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i^t - y_i^t log\hat{y}_i^t) \qquad (20)$$

5 EXPERIMENTS

In this section, we conduct extensive experiments to verify the effectiveness of our proposed M2M framework. Our experiments are designed to answer the following questions: *1) What is the performance of our proposed M2M compared with other state-of-art methods? 2) What is the influence of each of M2M's components? 3) What is the effect of key hyper-parameters, such as the length of temporal sequence, the dimension of scenario knowledge, the number of layers in meta unit? 4) What is the online A/B test performance of our proposed M2M?*

5.1 Experimental Settings

5.1.1 Dataset Statistics. We conducted experiments on a large-scale industrial dataset collected from Taobao, ranging from 2019-12-10 to 2020-12-30. There are 5 marketing scenarios including sponsored search, display ads, star shop ads, news feed ads, and brand search ads denoted from Sce.A to Sce.E, with a total of 1.83 million advertisers, and 30 million samples in the data set as shown in Table 3. As mentioned above, future PV(Page View or impressions), Click, Expenditure, GMV, Active rate are typical tasks that

our model predicts in the dataset. In this evaluation, we used previous 40-day features to predict the next 7-day performance, and partitioned the data based on the time axis into non-overlapping training, evaluation, and test data, by a ratio of 8:1:1. It is noteworthy that the interval between training set and testing set should be no less than one week in case of data leakage. The dataset's statistical information is exhibited in table 3.

5.1.2 Comparative Methods. We compare M2M with the following baselines:

- **Single-task model**. Use the Embedding&MLP architecture as the basis single-task deep learning model.
- **Shared Bottom**. The shared bottom model shares the bottom layer in multi-task learning.
- **Cross-Stitch Network**. Cross-Stitch[22] uses a linear cross-stitch unit to learn to optimally combine multiple network of representations for each task.
- **MMoE**. MMoE[20] makes use of a shared module named expert to learn the underlying feature representations. Each task uses its own gate network to combine experts to capture correlations and differences between the tasks.
- **MRAN**. MRAN[39] contains a task-feature dependence learning module, which can associate the related features with target tasks separately.
- **CGC**. CGC[32] consists of task-specific experts and the shared experts in representation layer, which avoids parameter conflicts resulting from complex task correlations.

5.1.3 Implementation Details. For fair comparison, we search for optimal parameters on validation data and evaluate these models on test data; Besides, scenario related information is also added as the input features to base models. In the shared bottom embedding layer of M2M framework, we set the hidden state dimension d_{input} as 16. In the multi-behavior transformer module, we set the number of attention heads for multi-dimensional learning as 2. To better characterize the relationship of features and tasks, we introduce feature views and task views representation, all feature views dimensions and task views dimensions are set to $d_{view} = 256$. We use LeakyReLU[21] as the activation function for all models. During training, the batch size is set to 256. We use Adam optimizer [16] with the setting $\beta 1 = 0.9, \beta 2 = 0.998$ and $\varepsilon = 1 \times 10^{-9}$. The learning rate is set to 2×10^{-3}. Gradient clipping is applied with range [-3, 3]. The model is trained in 10 epochs.

All the above models are trained using XDL[1] platform. The trained parameters are distributed on 40 workers (15 CPU cores for each worker) and updated asynchronously.

5.1.4 Evaluation Metrics. Commonly used metrics for the evaluation of regression problems are MAPE (Mean Absolute Percentage Error) which evaluates the micro average performance, and NMAE (Normalized Mean Absolute Error) which evaluates the macro average performance. In our tasks, the label of pv, clicks, etc, may be zero which would result in undefined value of MAPE. Therefore, we adopt SMAPE (Symmetric Mean Absolute Percentage Error),

---

[1]https://github.com/alibaba/x-deeplearning

Table 4: Comparison with State-of-the-arts. Sce.a represents Sponsor Search scenario; Sce.b represents Display scenario which is a <u>minor scenario</u>; Sce.c represents Star Shop scenario which is a <u>new scenario</u>. M2M outperforms all due to 2 key advantages: the well designed Meta Unit, and the effective meta learning mechanism that captures diverse inter-scenario correlations given different tasks.

Method	Task.click(Sce.a)		Task.click(Sce.b)		Task.click(Sce.c)		Task.active(Sce.a)		Task.active(Sce.b)		Task.active(Sce.c)		Overall(5Task*5Sce)	
	NMAE	SMAPE	NMAE	SMAPE	NMAE	SMAPE	NMAE	SMAPE	NMAE	SMAPE	NMAE	SMAPE	NMAE	SMAPE
Single Task	0.2398	0.4684	0.6687	0.8354	0.8205	0.8971	0.2603	0.2345	0.4146	0.3501	0.3653	0.3620	0.5372	0.6349
Shared Bottom	0.2740	0.4908	0.7486	0.8406	0.9388	0.9349	0.3112	0.2693	0.3897	0.3341	0.3246	0.3401	0.5561	0.6411
Cross-Stitch	0.2474	0.4993	0.6657	0.8241	0.7621	0.8902	0.3612	0.3112	0.3703	0.2617	0.2820	0.2745	0.4984	0.6189
MMoE	0.2449	0.4711	0.6869	0.8085	0.7306	0.8429	0.3195	0.2674	0.3631	0.2686	0.2711	0.2649	0.4953	0.6101
MRAN	0.2254	0.4566	0.6668	0.7797	0.7103	0.8185	0.3252	0.2686	0.3735	0.2707	0.2703	0.2654	0.4921	0.6068
CGC	0.2277	0.4765	0.6235	0.8043	0.7377	0.8351	0.2714	0.2578	0.3394	0.2721	0.2476	0.2550	0.4879	0.6087
M2M	**0.2168**	**0.4435**	**0.4733**	**0.7605**	**0.5406**	**0.7611**	**0.2354**	**0.2167**	**0.3063**	**0.2066**	**0.2158**	**0.2068**	**0.4109**	**0.5722**

(a) Overall Performance in NMAE   (b) Overall Performance in SMAPE

Figure 5: Overall Performance.

which is calculated as follows :

$$SMAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{|\hat{y}_i - y_i|}{(\hat{y}_i + y_i)/2} \tag{21}$$

And NMAE metric is calculated as follows:

$$NMAE = \frac{\sum_{i=1}^{N} |\hat{y}_i - y_i|}{\sum_{i=1}^{N} y_i} \tag{22}$$

where N is the number of advertisers, $\hat{y}_i$ and y_i are the predicted tasks and ground-truth of i-th advertiser, respectively.

5.2 Overall Performance Comparison

We show experiment results in Table 4, where we find that our M2M outperforms other baselines consistently on different tasks from different scenarios. This demonstrates the effectiveness of our proposed method. Specifically, For Task Click respectively in scenarios A, B and C, our model achieves (+4.7%, +24.1%, +26.7%) and (+6.9%, +5.4%,+8.9%) improvements over the best baseline on NMAE and SMAPE. Similarly, the improvement of NMAE and SMAPE on Task Active-rate are (+13.3%, +9.8%, +12.8%) and (+15.9%, +24.1%, +18.9%) respectively. Moreover, we find that MMoE, CGC, Cross-Stitch methods outperform the single task learning approach, which demonstrates the effectiveness of learning multiple tasks jointly by exploiting relationships between tasks. Interestingly, we notice that improvements to M2M on Task Click turn out to be quite different from scenario to scenario. This might be related to the fact that scenario A has a larger scale, while smaller-scale scenarios B and C record larger improvements as M2M has the ability to transfer scenario knowledge via meta unit.

Table 5: Results on the ablation study of the proposed M2M.

Model	Task.click(Sce.a)		Task.click(Sce.b)		Task.click(Sce.c)	
	NMAE	SMAPE	NMAE	SMAPE	NMAE	SMAPE
Full Model	**0.2168**	**0.4435**	**0.4733**	**0.7605**	**0.5406**	**0.7611**
w/o MT	0.2241	0.4612	0.5710	0.7923	0.6593	0.8158
w/o MA	0.2189	0.4461	0.4966	0.7883	0.5768	0.7836
w/o TL	0.2200	0.4493	0.5282	0.7612	0.5465	0.7643

Finally, the detailed performance is displayed in Figure 5(a) and Figure 5(b), where we use NMAE and SMAPE to evaluate the overall performance of M2M with the best baseline CGC. Our M2M significantly outperforms CGC consistently on multi-scenario and multi-task.

5.3 In-depth Analysis

To better understand the performance of the proposed model, we conduct a series of in-depth analysis on M2M.

5.3.1 Ablation Analysis. To validate the contribution of each component of M2M, we conduct an ablation study by examining the performance after removing each component, listed and denoted as follows.

- **M2M(Full Model)**: Use all components.
- **w/o meta tower module (MT)**. Simple version without learning the high-order interactive features for a specific scenario.
- **w/o meta attention module (MA)**. Simple version without capturing diverse inter-scenario correlations.
- **w/o transformer Layer (TL)**. Simple version without transformer layer.

Table 5 shows the effectiveness of each module. And results are summarized as following: (1) The meta learning mechanism which contains meta attention module and meta tower module plays an important role in M2M. As shown, M2M without meta tower module and M2M without meta attention module are inferior to M2M. This proves that the meta attention module and meta tower module are effective in identifying lower-level fusion patterns and higher-level mixture patterns for specific scenarios respectively.

(a) Temporal Length T   (b) Temporal Length T   (c) Scenario Knowledge Dimension d (d) Scenario Knowledge Dimension d   (e) Meta Unit Depth K   (f) Meta Unit Depth K
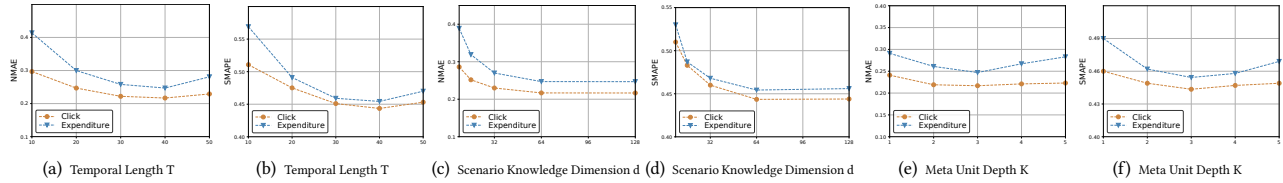
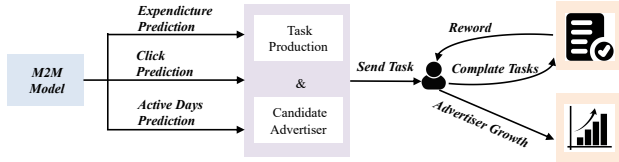Figure 6: Hyperparameter Sensitivity.

Figure 7: The pipeline of advertiser modelling and rewarded task tool application.

(2) As shown, M2M without transformer layer is inferior to M2M. This proves that transformer layer can effectively model the the time-series dependency.

5.3.2 Hyperparameter sensitivity. In the following studies, we learn the impact of key hyper-parameters including the temporal sequence length T, the dimension of scenario knowledge d, and the number of neural network layers K in our meta unit one at a time by holding the remaining hyper-parameters at the optimal settings. The evaluation results in predicting click and expenditure tasks with SMAPE and NMAE metrics are shown in Figure 6. The major findings are summarized as follows :

Figure 6(a) and figure 6(b) presents the results of NMAE and SMAPE with different temporal sequence length T, respectively. The metrics reach optimal when the sequence length is 40 and the performance decreases when the sequence becomes longer. This can be explained by the reason that a longer sequence length tends to introduce extra information and more noises. The dimension of scenario knowledge d is another important parameter. To check the sensitivity of d, we plot overall NMAE and SMAPE of M2M with d = 8, 16, 32, 64 and 128 in Figure 6(c) and 6(d), with the increase of d, the model gets better performance. After d increases to 64, the result of model prediction is relatively stable. Figure 6(e) and 6(f) show the NMAE and SMAPE for the different numbers of layers K in the meta unit. From this figure, we find that deeper networks can not guarantee to achieve better results. The model achieves its best performance when K is 3. When the network becomes deeper, the performance deteriorates. This is due to the reason that a deeper network is more likely to suffer from overfitting issues.

5.4 Online A/B Test

The proposed M2M method for advertisers modeling builds the foundation for many downstream applications such as Rewarded Task and Push Message which aim to improve the engagement and retention of advertisers in Taobao ecosystem. As Figure 7 demonstrates, with our M2M model, we obtain the predictions of expenditure, clicks and active days in next 7 days, and then select the candidate advertisers whose future active rate is low; Next, the Task Production module assigns personalized tasks on top of those

Table 6: A/B Test of the proposed model compared to the base model.

Indicators	Gain
Active Rate	+2.59%
Average Revenue Per User	+2.09%

predictions. Selected advertisers are motivated to accomplish these tasks in order to get rewards.

Table 6 presents the results of online AB tests on the Rewarded Task Tool. Between 2021-March-11 and 2021-March-29, we conducted several rounds of online A/B tests in different scenarios. Take sponsored search scenario for example, the new M2M model and Single Task base model generated equal size of predicted future-inactive advertisers respectively as test and base group, about 100,000 each. With the same type of incentive tasks and incentive budget applied, we observed an improvement of +2.59% in terms of the active rate, +2.09% in terms of ARPU(Average Revenue Per User). This online experiment demonstrates the effectiveness and practicability of our model in the industrial landscape.

6 CONCLUSION

In this paper, we proposed an efficient and scalable framework named M2M for multiple-scenario multiple-task advertiser modeling. First, a backbone network is adopted to learn representations of advertiser-related features and tasks. Then, to learn the meta-knowledge of scenario-related information explicitly, we design a Multi-level Meta Unit with customized scenario-related input, and its output is used to generate layer weights for later prediction networks. Furthermore, to combine multiple-scenario multiple-task in an end-to-end fashion effectively, we propose a Meta Learning Mechanism, which comprises a meta attention module and a meta tower module, respectively to capture diverse scenario-task correlations and enhance scenario-specific feature representation. Finally, we utilize the joint Poisson loss for multiple tasks to optimize our model in an end-to-end manner. Compelling results from both offline evaluation and online A/B tests demonstrate the superiority of M2M over state-of-the-art methods, especially in new and minor marketing scenarios. For future work, we'd like to incorporate more types of tasks like classifier beyond regression on top of the prediction network. Besides, it might also be interesting to combine our M2M model with optimization-based meta learning approaches like MAML, Reptile etc.

REFERENCES

[1] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip HS Torr, and Andrea Vedaldi. 2016. Learning feed-forward one-shot learners. *arXiv preprint arXiv:1606.05233* 14 (2016).

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

[3] Collobert, Ronan, Weston, and Jason. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008.*

[4] Li Deng, Geoffrey Hinton, and Brian Kingsbury. 2013. New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE international conference on acoustics, speech and signal processing.* IEEE.

[5] Theodoros Evgeniou, Charles A Micchelli, Massimiliano Pontil, and John Shawe-Taylor. 2005. Learning multiple tasks with kernel methods. *Journal of machine learning research* 6, 4 (2005).

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning.* PMLR, 1126–1135.

[7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[8] Liyi Guo, Rui Lu, Haoqi Zhang, Junqi Jin, Zhenzhe Zheng, Fan Wu, Jin Li, Haiyang Xu, Han Li, Wenkai Lu, et al. 2020. A Deep Prediction Network for Understanding Advertiser Intent and Satisfaction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 2501–2508.

[9] David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106* 15 (2016).

[10] Jindong Han, Yuan He, Juan Liu, Qianqian Zhang, and Xiaojun Jing. 2019. Graph-ConvLSTM: Spatiotemporal Learning for Activity Recognition with Wearable Sensors. In *2019 IEEE Global Communications Conference (GLOBECOM).* IEEE, 1–6.

[11] Jindong Han, Hao Liu, Hengshu Zhu, Hui Xiong, and Dejing Dou. 2021. Joint Air Quality and Weather Prediction Based on Multi-Adversarial Spatiotemporal Networks. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence.*

[12] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[14] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439* (2020).

[15] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 7482–7491.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

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

[18] Hao Liu, Jindong Han, Yanjie Fu, Jingbo Zhou, Xinjiang Lu, and Hui Xiong. 2020. Multi-modal transportation recommendation with unified route representation learning. *Proceedings of the VLDB Endowment* 14, 3 (2020), 342–350.

[19] Hao Liu, Yongxin Tong, Jindong Han, Panpan Zhang, Xinjiang Lu, and Hui Xiong. 2020. Incorporating multi-source urban data for personalized and context-aware multi-modal transportation recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[20] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1930–1939.

[21] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. Citeseer, 3.

[22] 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.* 3994–4003.

[23] Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *International Conference on Machine Learning.* PMLR, 2554–2563.

[24] A. Nichol, J. Achiam, and J. Schulman. 2018. On First-Order Meta-Learning Algorithms. (2018).

[25] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1720–1730.

[26] Huimin Peng. 2020. A Comprehensive Overview and Survey of Recent Advances in Meta-Learning. *arXiv preprint arXiv:2004.11149* (2020).

[27] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2671–2679.

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

[29] G. Pironkov, S. Dupont, and T. Dutoit. 2016. Multi-Task Learning for Speech Recognition: An Overview. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN).*

[30] S. Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. (2017).

[31] Jürgen Schmidhuber. 2007. Gödel machines: Fully self-referential optimal universal self-improvers. In *Artificial general intelligence.* Springer, 199–226.

[32] Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).* 231–235.

[33] 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 *Fourteenth ACM Conference on Recommender Systems.* 269–278.

[34] Sebastian Thrun and Lorien Pratt. 2012. *Learning to learn.* Number 1. Springer Science & Business Media.

[35] Joaquin Vanschoren. 2018. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548* (2018).

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention Is All You Need. *arXiv* (2017).

[37] Sangho Yoon, Jim Koehler, and Adam Ghobarah. 2010. Prediction of advertiser churn for google adwords. 18 (2010).

[38] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017).

[39] Jiejie Zhao, Bowen Du, Leilei Sun, Fuzhen Zhuang, Weifeng Lv, and Hui Xiong. 2019. Multiple relational attention network for multi-task learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1123–1131.

[40] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[41] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1059–1068.