Recurrent Meta-Learning against Generalized Cold-Start Problem in CTR Prediction

Junyu Chen
SKLOIS, IIE, CAS
SCS, UCAS
chenjunyu@iie.ac.cn

Qianqian Xu*
IIP, ICT, CAS
xuqianqian@ict.ac.cn

Zhiyong Yang
SCST, UCAS
yangzhiyong21@ucas.ac.cn

Ke Ma
SCST, UCAS
make@ucas.ac.cn

Xiaochun Cao
SCST, Shenzhen Campus, SYSU
SKLOIS, IIE, CAS
caoxiaochun@mail.sysu.edu.cn

Qingming Huang*
SCST, UCAS
IIP, ICT, CAS
BDKM, CAS
Peng Cheng Laboratory
qmhuang@ucas.ac.cn

ABSTRACT

During the last decades, great success has been witnessed along the course of accurate Click-Through-Rate (CTR) prediction models for online advertising. However, the cold-start problem, which refers to the issue that the standard models can hardly draw accurate inferences for unseen users/ads, is still yet to be fully understood. Most recently, some related studies have been proposed to tackle this problem with only the new users/ads being considered. We argue that such new users/ads are not the only sources for cold-start. From another perspective, since users might shift their interests over time, one's recent behaviors might vary greatly from the records long ago. In this sense, we believe that the cold-start problem should also exist along the temporal dimension. Motivated by this, a generalized definition of the cold-start problem is provided where both new users/ads and recent behavioral data from known users are considered. To attack this problem, we propose a recursive meta-learning model with the user's behavior sequence prediction as a separate training task. Specifically, a time-series CTR model with the MAML (Model-Agnostic Meta-Learning)-like meta-learning method is proposed to make our model adapt to new tasks rapidly. Besides, we propose a parallel structure for extracting the feature interactions to efficiently fuse attention mechanisms and the RNN layer. Finally, experiments on three public datasets demonstrate the effectiveness of the proposed approaches.

CCS CONCEPTS

• **Information systems** → **Online advertising**.

KEYWORDS

Click-Through Rate Prediction, Meta-Learning, Cold-Start Problem

MM '22, October 10–14, 2022, Lisboa, Portugal.

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548118

ACM Reference Format:
Junyu Chen, Qianqian Xu, Zhiyong Yang, Ke Ma, Xiaochun Cao, and Qingming Huang. 2022. Recurrent Meta-Learning against Generalized Cold-Start Problem in CTR Prediction. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), Oct. 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548118

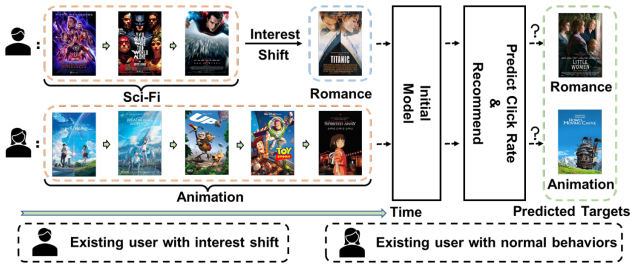


Figure 1: An illustration of interest shift in CTR. For example, the first user's historical interaction shifts from science fiction movies to romance movies, so the historical science fiction movies do not help much to predict the romance movie next time; the second user has been watching cartoons, and these interactions contribute to the prediction of the next cartoon viewing.

1 INTRODUCTION

Click-Through Rate (CTR) prediction, which aims to predict the possibility that a user clicks an item, has been playing an essential role in many applications, e.g., e-commerce and media sites. Seeing the importance of CTR, a substantial number of studies have been carried out in search of accurate prediction for CTR [2, 7, 10, 14, 32, 40, 44, 45]. For the CTR prediction task, one crucial challenge is the cold-start problem, which requires the models to give accurate predictions for unseen elements such as new users and new ads. A large number of researchers have studied this problem by utilizing auxiliary information [19, 26, 38, 42], utilizing limited interactions more effectively [18, 23] or introducing exploit & exploration techniques [20, 25], etc.

*Corresponding authors.

However, since online platforms and users usually have complicated dynamical properties, the cold-start problem should as well be understood from multiple dimensions. From one dimension, predictions of new users' preferences are naturally seen as tasks in the cold start problem. Nonetheless, this doesn't mean that the old users should be ignored during this process. In fact, when personalized interests change gradually over time, one's new behaviors might vary greatly from the records long ago, as shown in Figure 1. To obtain this new hidden information, we need to fine-tune the model to make better predictions. In this sense, the second dimension of the cold-start problem is then to predict new sequences of CTR results from old users which either have partial relation with the historical dataset or are completely independent of previous records.

Based on the spirits above, we provide a generic definition of the cold-start problem in this paper and consider both new users and the new behaviors of old users. We regard the prediction of a partial sequence (called **session** hereafter), as a single task to tackle the generalized cold-start in a unified way. This setting shares a significant resemblance with the meta-learning problem which intends to design models that can adapt to new environments rapidly with a few training examples. Inspired by this, we propose a recursive meta-learning model. Specifically, there are two key designs in our model: (a) We propose a meta-interaction module to effectively exploit the dynamic nature of sessions. It is a parallel architecture consisting of an LSTM branch and an attention branch, where the LSTM branch is to extract the evolution of interests and the attention branch learns from correlation with target ads to better predict the click rate. (b) We propose a novel meta-learning training strategy, where two training subroutines are employed to optimize the model parameters. The first subroutine, i.e., **the local training**, updates task-specific parameters to adapt to individual tasks, while the second subroutine, i.e., **the global training**, initializes all parameters to learn proper parameters for all tasks.

In short, the main contributions of this work are three-fold:

- We provide a more generalized definition of the cold-start problem against new users and new records of old users. Besides, a new setting of the meta-learning method is proposed to solve cold-start problems, which considers user behavior sequence prediction instead of a single user as a separate task.
- We propose a sequential model based on user behavior sequences to learn how to learn a better meta-interactions layer, which is a parallel composite structure of the attention mechanism and the LSTM layer, through recurrent meta-learning.
- A novel two-phase meta-learning training strategy is designed to mitigate the generalized cold-start problem.

We conduct experiments on three real-world datasets with some state-of-the-art CTR models. Moreover, ablation studies are also conducted on the proposed designs. Experimental results speak to the effectiveness of our method.

2 RELATED WORK

2.1 General CTR Prediction

For general CTR predictions, early researches treat users' historical behaviors independently and equally important, representative work including traditional collaborative filtering methods, such as [2, 10, 11, 16, 21, 28, 29, 32, 39], etc. Lately, some attention-based models [34, 45] are developed to assign different items different weights when predicting users' next behavior. Remarkable success as they made, these methods ignore the sequential information of historical behaviors and fail to infer the evolution of users' interests. To this end, some methods [1, 4, 6, 12, 27, 31] turn to sequential modules such as LSTM [13], GRU, memory networks, etc. For example, in MIMN [27], there is a memory induction unit that uses a multi-channel GRU to store long sequence information. This line of research can reveal the evolution of user preferences, and thus may help make predictions according to users' current interests. The above two lines of methods have their own focuses, lately, some methods propose to combine both advantages. For example, DIEN [44] applies an attention mechanism to the intermediate memories of sequential modules such that memories at different moments contribute differently to the final prediction. DSIN [7] divides the user's behavior sequence into sessions. On this basis, it uses a biased self-attention network to capture the correlation between sessions and get the session interest expression.

However, faced with interest drifting, the above combinations may be insufficient to exclude the inconsistent information from noisy historical items. This is because that the memories extracted by sequential modules at each moment are affected by the previous moments, which makes the evolution of learned memories susceptible to historical interference. If attention mechanisms are applied to the memories, the historical inconsistent information may be even amplified. Motivated by this limitation, this paper proposes a parallel composite structure of sequential modules and attention layers to extract the evolution of interests and target correlation, respectively. This parallel structure helps to avoid mutual propagation of inconsistency and compensate each other.

2.2 Meta-Learning

Meta-learning enables models to adapt to new training tasks via learning how to learn [3, 8, 15, 35, 37, 41]. When considering specific methods, there are lots of applications: hyper-parameter optimization [9], neural architecture search (NAS) [22, 30], few-shot learning [17], reinforcement learning [46], etc.

One of the current mainstream meta-learning methods is known as Model-Agnostic Meta-Learning (MAML) [8]. The idea of it is to train the initial parameters of the model by a special gradient update strategy. So that when facing a new task with only a small amount of data, it only needs to go through a few gradient updates steps to fine-tune the parameters with better performance. Another popular method is known as Reptile [24], which is developed from MAML. Its training process is divided into inner and outer loops like MAML. However, Reptile does not need to calculate the second-order gradient.

In recent years, due to the similarity between meta-learning and the cold-start problem of recommendation systems, it is applied in recommendation systems [18, 26, 36]. Meta-Embedding [26] uses

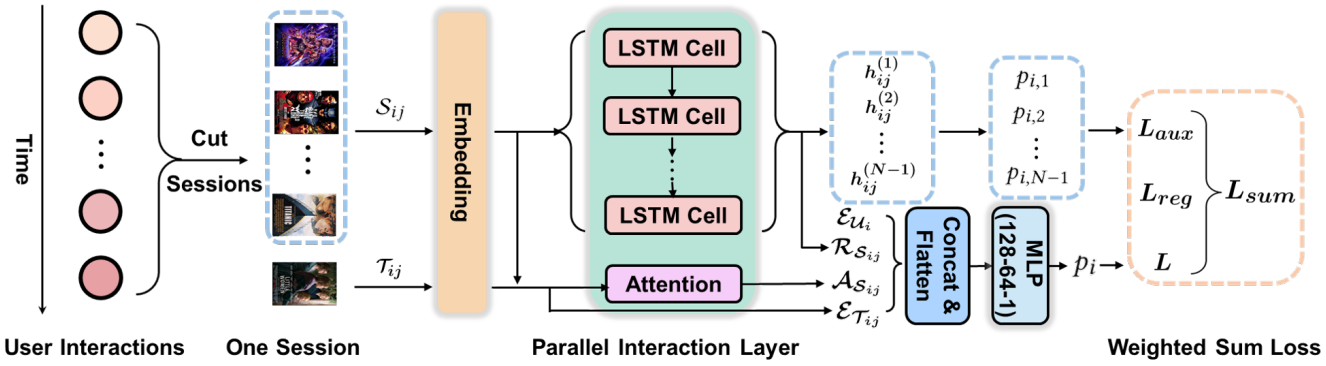


Figure 2: The pipeline of the proposed methods. First, we split the sequence of user behaviors into different sessions. For each session $\mathcal{S}_{ij}$ with the predicted target $\mathcal{T}_{ij}$, we transform their discrete values into concatenated matrices formed by different embedding vectors: $\mathcal{E}_{\mathcal{S}_{ij}}$ and $\mathcal{E}_{\mathcal{T}_{ij}}$. Then they will enter the parallel interaction layer, which is the parallel combination of LSTM and the attention layers. The output of the parallel interaction layer $(\mathcal{R}_{\mathcal{S}_{ij}}, \mathcal{A}_{\mathcal{S}_{ij}})$, the target item embedding $\mathcal{E}_{\mathcal{T}_{ij}}$ and user embedding $\mathcal{E}_{\mathcal{U}_i}$ are concatenated and flattened to be the input of the last MLP part to get the probability p_{ij}.

meta-learning to learn the embedding vector representation of new ads which has never been encountered by a fully connected layer. It regards each Ad as a task and uses the existing data to train the embedding generator. MeLU [18] is another example with a user-oriented idea. It divides the dataset into old users and new users, and uses the data of old users to train the user preference estimator in a meta-learning method. Besides, MAMO [5] contains a feature-specific memory to provide a personalized bias term when initializing the model parameters. A task-specific memory cube also is designed to learn to capture the shared potential user preference commonality on different items.

In this paper, our method is quite different from these. We regard the prediction of one session as a task, and use two training subroutines to learn task-specific and task-shared parameters respectively in the meta-learning method.

3 METHODOLOGY

In this section, we will elaborate on our model from the following aspects. First, we give the definition of the generalized cold start problem and introduce the basic setting. Then we elaborate our model structure. After that, we introduce our customized meta-learning strategy.

3.1 Problem Setup

As mentioned before, we argue that the cold-start problem should also include the dynamic shift of the users' interests. Before entering into our methodology, we first provide a formal definition of this problem.

Problem 1 (**Generalized Cold-Start**). *The Generalized Cold-Start problem refers to the challenge of simultaneously predicting CTR values for both (a) unseen users and (b) the new reactions from known users.*

To achieve (a), one must find efficient ways to adapt to new users rapidly. And for (b), one shall correctly model the relations across different time periods. Motivated by this, we regard the records prediction from a specific time region and a specific user as an individual task. Precisely, we divide a specific user i's original historical behavior sequence $\mathcal{H}_i$ into different sessions according to a certain time interval: $\mathcal{H}_i = \{S_{i1}, S_{i2}, \cdots, S_{iN_i}\}$, where N_i is the number of sessions for the i-th user. We call the raw feature of the j-th session as a set of item features from a given region, i.e.,

$$\mathcal{S}_{ij} = \left\{\mathcal{I}_{ij}^{(1)}, \mathcal{I}_{ij}^{(2)}, \cdots, \mathcal{I}_{ij}^{(L)}\right\},$$

where $\mathcal{I}_{ij}^{k}$ is the features of the k-th item, L is the maximum length of one session.

We regard the prediction of user i's j-th session as an individual task (i, j). Denote the input data of the j-th session for user i as $\mathcal{D}_{ij}$, and $\mathcal{D}_{ij} = \{\mathcal{U}_i, \mathcal{S}_{ij}, \mathcal{T}_{ij}, y_{ij}\}$. Herein, $\mathcal{U}_i$ is user i's features (*eg., age and gender*), $\mathcal{T}_{ij}$ is the features of the target item for $\mathcal{S}_{ij}$ and y_{ij} represents the label of the target item.

3.2 Model Design

Our model includes the following building blocks: the embedding layer, the parallel interaction layer, and the MLP layers. The whole framework is also shown in Figure 2.

3.2.1 Embedding Layer. In the CTR prediction, data is usually collected in a multi-field categorical form [33, 43]. So we first adopt an embedding layer to convert discrete features into a dense low-dimensional space:

$$\begin{aligned}\mathcal{E}_{ij} &= \text{LookUp}\left(\{\mathcal{U}_i, \mathcal{S}_{ij}, \mathcal{T}_{ij}\}\right) \\ &= \text{LookUp}\left(\mathcal{U}_i\right) \oplus \text{LookUp}\left(\mathcal{S}_{ij}\right) \oplus \text{LookUp}\left(\mathcal{T}_{ij}\right) \\ &= \mathcal{E}_{\mathcal{U}_i} \oplus \mathcal{E}_{\mathcal{S}_{ij}} \oplus \mathcal{E}_{\mathcal{T}_{ij}}\end{aligned} \tag{1}$$

where $\oplus$ means the concatenating operation, $\mathcal{E}_{\mathcal{U}_i}, \mathcal{E}_{\mathcal{S}_{ij}}$, and $\mathcal{E}_{\mathcal{T}_{ij}}$ are the embedded features of user i, session (i, j) and its target item, respectively.

3.2.2 Meta-Interaction Module. On top of the embedded features, we propose the meta-interactions layer, which is the foundation of the meta-learning strategy presented later. In this layer, we employ

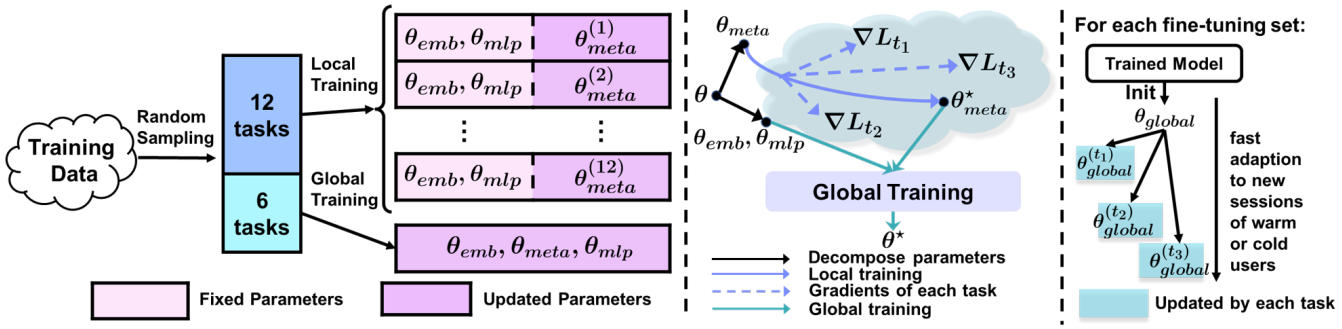


Figure 3: Our meta-learning strategy for one batch. Each batch of tasks is divided into a support set (here the former 12 tasks) and a query set (here the latter 6 tasks). During the local training, the support tasks are utilized to update θ_{meta} while the remaining parameters are fixed. During the global training, we update all parameters θ with the query set. The middle sub-figure shows the relationship between local training and global training more specifically. The right sub-figure shows our setting during the testing phase.

a parallel composite structure of the attention mechanism and the RNN layer to extract features. The attention mechanism focuses on historical information that weighs heavily on the current forecast item. And as for the RNN layer, it mainly finds the latent features, which may influence the user's choice, like hidden relations between items or the user's interests. After that, the resulting output of both is fused to generate the final result. The parallel structure is composed of two branches: the attention branch and the LSTM branch.

For the attention branch, we first calculate the attention weights of each item in $\mathcal{S}_{ij}$ w.r.t the target item $\mathcal{T}_{ij}$. Then we weigh items in the session to obtain the aggregated session representations $\mathcal{A}_{\mathcal{S}_{ij}}$ w.r.t the target item. These two procedures can be represented as:

$$\mathcal{A}_{\mathcal{S}_{ij}} = \mathcal{E}_{\mathcal{S}_{ij}} \cdot \text{ffn}\left(\mathcal{E}_{\mathcal{S}_{ij}}, \mathcal{E}_{\mathcal{T}_{ij}}\right) \tag{2}$$

where ffn $(\cdot)$ is the feed-forward network that outputs the attention weights of each item w.r.t the target item.

At the same time, for the LSTM branch, we obtain the latent representation of these items' sequential relationship by the LSTM modules:

$$\mathcal{R}_{\mathcal{S}_{ij}} = \text{LSTM}\left(\mathcal{E}_{\mathcal{S}_{ij}}\right) \tag{3}$$

The final output of this layer is then the concatenation and flattening of features for both branches, together with some other context features. Formally, we have:

$$C_{ij} = \text{Concat\&Flatten}\left(\mathcal{A}_{\mathcal{S}_{ij}}, \mathcal{R}_{\mathcal{S}_{ij}}, \mathcal{E}_{\mathcal{U}_i}, \mathcal{E}_{\mathcal{T}_{ij}}\right) \tag{4}$$

where C_{ij} is the final features that will be fed to MLP layers.

Discussion. There have been some related works that combined the advantages of both the attention mechanism and the sequential modules. DIEN [44] applies an attention mechanism to the intermediate memories of LSTM such that memories at different moments contribute differently to the final prediction. However, faced with interest drifting, this kind of combination may be insufficient to exclude the inconsistent information from noisy historical items. This is because that the memories extracted by sequential modules at each moment are affected by the previous moments, which makes the evolution of learned memories susceptible to historical interference. If attention mechanisms are applied to the memories, the historical inconsistent information may be even amplified. Our proposed meta-interaction modules employ a parallel composite of sequential modules and attention layers to extract the evolution of interests and target correlation, respectively. This parallel structure helps to avoid mutual propagation of inconsistency and compensate each other. Hence, it can exploit the session information more effectively and help adapt to new sessions rapidly.

3.2.3 MLP Layers. Based on the final features for each task (i, j), we employ an MLP layer to predict the click rate. For the activation function, we use Dice [45] in the intermediate layers. For the last fully connected layer, *i.e.*, the output layer, we adopt the sigmoid function $\sigma(\cdot)$ to restrain the output probability to the interval $[0, 1]$. The whole process can be shown below:

$$p_{ij} = \sigma(\text{MLP}\left(C_{ij}\right)) \tag{5}$$

where p_{ij} is the predicted probability that user i during session j will click the target item $\mathcal{T}_{ij}$.

3.2.4 Loss Functions. For each task (i, j), given the predicted click rate p_{ij} and the ground truth label of the target item y_{ij}, we adopt the negative logarithmic likelihood function as the loss function, i.e.,

$$L^{(i,j)} = -y_{ij} \log p_{ij} - \left(1 - y_{ij}\right) \log\left(1 - p_{ij}\right) \tag{6}$$

where y_{ij} is the corresponding ground-truth label.

In addition, to supervise the intermediate hidden states in LSTM as well, we employ an auxiliary loss. Denote the hidden state of step $t-1$ in LSTM as h^{t-1}_{ij}, and the representation of the item at step t as $\mathcal{E}_{\mathcal{I}^t_{ij}}$. We use the item at step t to supervise the learning process of h^{t-1}_{ij}. Then we could get the auxiliary loss:

$$L^{(i,j)}_{aux} = -\sum_{t=2}^{L} \left(y_{\mathcal{I}^t_{ij}} \log p^t_{ij} + \left(1 - y_{\mathcal{I}^t_{ij}}\right) \log\left(1 - p^t_{ij}\right)\right) \tag{7}$$

where $p^t_{ij} = \sigma\left(h^{t-1}_{ij}, \mathcal{E}_{\mathcal{I}^t_{ij}}\right)$ and $y_{\mathcal{I}^t_{ij}}$ are the predicted CTR and ground truth label of the item at step t, respectively.

Besides, we use ℓ_2 regularization for the attention layer and MLP and obtain the regularization loss L_{reg}. Finally, the total loss is made

up of all the loss functions:

$$L_{sum}^{(i,j)} = \alpha \cdot L^{(i,j)} + \beta \cdot L_{reg} + \gamma \cdot L_{aux}^{(i,j)} \tag{8}$$

where the hyper-parameters α, β, γ balance the importance of these losses.

3.3 Meta-Learning Strategy

The proposed meta-learning strategy includes two training subroutines: the local training and the global training, as shown in Figure 3. Since the meta-interaction layers adaptively learn users' interests in different sessions, we update the parameters of the meta-interaction layers during the local training. Assuming that the user and item's embeddings do not change, we do not update the embedding layers and the MLP layers during the local training to ensure the stability of the learning process. Denote the parameters of the embedding layers, meta-interaction layers and the MLP layers as $\theta_{emb}, \theta_{meta}$ and θ_{mlp}, respectively. For each batch of sessions/tasks, we sample B tasks, where the first $B - l_{pos}$ tasks serve as the support set during local training, while the remaining l_{pos} tasks serve as the query set during global training, where $0 < l_{pos} \leq B$ is a hyper-parameter.

Local Training. During the local training, we fix θ_{emb} and θ_{mlp} and update θ_{meta} with the first $B - l_{pos}$ tasks:

$$\theta_{meta} \leftarrow \theta_{meta} - \eta_1 \sum_{(i,j)} \frac{1}{B - l_{pos}} \nabla_{\theta_{meta}} L_{sum}^{(i,j)} \tag{9}$$

where η_1 is the learning rate of the local training. This process is the rapid adaption process from the initialized parameter to specific parameters that fit new sessions.

Global Training. During the global training, we use the remaining l_{pos} tasks to update all model parameters $\theta = \{\theta_{emb}, \theta_{meta}, \theta_{mlp}\}$:

$$\theta \leftarrow \theta - \eta_2 \sum_{(i,j)} \frac{1}{l_{pos}} \nabla_{\theta} L_{sum}^{(i,j)} \tag{10}$$

where η_2 is the learning rate of global training. This process aims to find desirable parameters that could achieve good performance for all tasks after several local updates, i.e., to find a proper parameter initialization for the future new sessions.

Fast Adaption. In the testing phase, to see how our model could adapt to new sessions, we use three fine-tuning subsets, called FT-A, FT-B and FT-C, respectively, to fine-tune the model. As shown in the right of Figure 3, we update the model parameters θ on each fine-tuning subset sequentially, and then validate our model on the testing set.

4 EXPERIMENTS

In this section, we evaluate the quantitative results of our method against other state-of-the-art baselines on three popular real-world datasets. All experiments are run on a machine with E5-2620 CPU, TITAN RTX and 256G RAM.

4.1 Benchmark Datasets

In this paper, we conduct experiments on the following real-world datasets:

Table 1: Statistics of the datasets. The sparsity is calculated by dividing the number of samples by the product of the users' number and the items' number.

	MovieLens	Book-Crossing	Avazu
#Users	6,040	278,858	2,686,408
#Items	3,900	271,379	40,428,967
#Samples	1,000,209	1,149,780	40,428,967
Sparsity	4.24609%	0.00152%	0.00004%

- **MovieLens-1M**[*]: It describes a 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 1 million ratings of approximately 3,900 movies made by 6,040 users who joined MovieLens in 2000.
- **Book-Crossing**[†]: It is collected from the Book-Crossing community, and contains 278,858 users providing 1,149,780 ratings (explicit/implicit) of about 271,379 books.
- **Avazu**[‡]: It was published in the contest of Avazu Click-Through Rate Prediction, 2014. This dataset provides 11 days of click-through data, including 40,428,967 ads with related feature fields.

The statistical information of our experimental datasets is shown in Table 1.

4.2 General Setting

The maximum length of a session is set to 10. Since we need to split user interactions into at least training and testing sessions, for each dataset, we filter out the users with less than 20 interactions, then divide the remaining users into warm/cold users at a ratio of 4/1. The items are classified as warm items if the number of their interactions is greater than a certain threshold, otherwise they are classified as cold items. The ratio of the two types of items is 1/1. Then, users and items are divided into two disjoint subsets, namely, **WC** (**w**arm users with **c**old items) and **CW** (**c**old users with **w**arm items). For each subset, the user's interactions are divided into several sessions in chronological order through a sliding window, and the last item of each session is regarded as the target item. For each user, all but the last session is regarded as the training sessions while the last session is randomly deployed into FT-A, FT-B, FT-C, or the testing set.

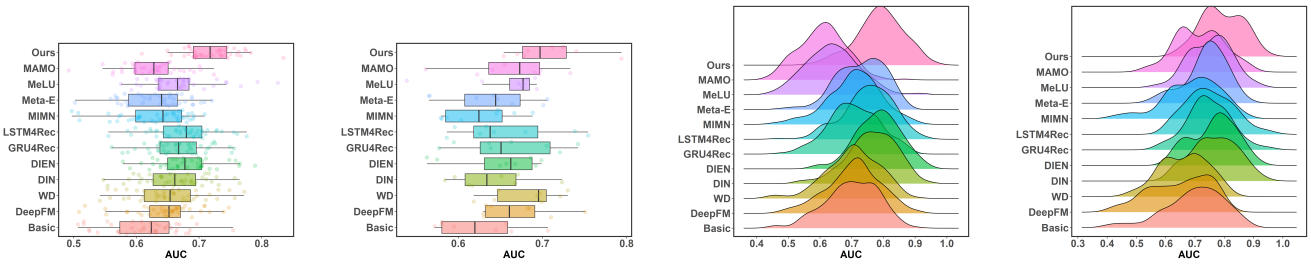
In the testing phase, we first fine-tune the trained model with FT-A, and get the results on the testing set. Then, we fine-tune the model on FT-B and FT-C successively, and evaluate the testing set. Due to space limitations, in the main paper, we only report the experimental results after fine-tuning with FT-C. Those results after fine-tuning on FT-A and FT-B are recorded in the Appendix.

We set the batch size to 128, and the maximum number of epochs varies with the model. During the meta-learning process, we set l_{pos} to 1/3 of batch size to balance the local and global training. We use the Adam optimizer with the exponential decay rate being 0.70. On MovieLens-1M, the initial learning rate is set to $\eta_1 = \eta_2 = 0.01$. We set $\alpha = \beta = 0.7$ and $\gamma = 0.3$. On the Book-Crossing dataset,

[*]https://grouplens.org/datasets/movielens/1m/
[†]http://www2.informatik.uni-freiburg.de/ cziegler/BX/
[‡]https://www.kaggle.com/c/avazu-ctr-prediction

Table 2: AUC and loss on MovieLens-1M dataset after fine-tuning with FT-A, FT-B and FT-C. The best results are marked red while the second-best is marked blue.

	FT-A				FT-B				FT-C			
	WC		CW		WC		CW		WC		CW	
	loss	AUC	loss	AUC	loss	AUC	loss	AUC	loss	AUC	loss	AUC
Basic	0.6041	0.6797	0.4726	0.6757	0.6118	0.6797	0.4739	0.6756	0.5998	0.6797	0.4755	0.6756
WD	0.5989	0.7041	0.4679	0.7423	0.6082	0.7212	0.4753	0.7183	0.5842	0.7350	0.4670	0.7315
DeepFM	1.7490	0.7231	1.6585	0.6725	1.7330	0.7239	1.7252	0.6692	1.6424	0.7233	1.6522	0.6833
DIN	1.8522	0.7671	0.6998	0.7804	1.9922	0.7676	0.6550	0.7959	1.8728	0.7588	0.6547	0.7895
DIEN	3.8806	0.7155	4.2122	0.5689	3.9700	0.7170	4.1354	0.6289	3.9747	0.7048	4.3587	0.6124
GRU4Rec	2.9831	0.7219	1.6285	0.7372	2.9831	0.7249	1.5997	0.7385	2.9051	0.7172	1.6766	0.7422
LSTM4Rec	2.6343	0.7842	1.2620	0.7718	2.3866	0.7812	1.2463	0.7761	2.4063	0.7859	1.3125	0.7685
MIMN	3.0208	0.7381	0.5514	0.5848	2.9719	0.7350	0.4650	0.7104	3.0135	0.7399	0.4362	0.7414
Meta-E	1.1852	0.7668	0.7710	0.8013	1.1603	0.7667	0.7777	0.8013	1.0702	0.7668	0.7709	0.8013
MeLU	2.5886	0.7677	1.4001	0.7653	2.5156	0.7682	1.4053	0.7702	2.3398	0.7666	1.4132	0.7661
MAMO	0.5928	0.6934	0.4906	0.6575	0.6068	0.6949	0.4884	0.6550	0.5969	0.6993	0.4863	0.6596
Ours	0.4642	0.8316	0.3454	0.8054	0.4881	0.8013	0.3253	0.8326	0.4730	0.8167	0.3001	0.8480



(a) AUC on WC of Book-Crossing (b) AUC on CW of Book-Crossing (c) AUC on WC of Avazu (d) AUC on CW of Avazu

Figure 4: AUC on Book-Crossing and Avazu after fine-tuning with the last FT-C. We draw the AUC values of each batch on the test data in various ways, boxplots for Book-Crossing and ridgelines for Avazu, respectively. Both kinds of plots generally show the distribution of AUC, which illustrates the convergence. Our proposed methods perform best compared with other competitors in the four sub-graphs.

the initial learning rate is set to 0.0001. The remaining settings are the same as those on MovieLens-1M. On Avazu, we set the initial learning rate of the WC subset and CW subset to 0.01 and 0.001, respectively. In addition, $\{\alpha, \beta, \gamma\}$ are set to $\{0.3, 0.3, 0.7\}$ for WC and $\{0.4, 0.4, 0.6\}$ for CW, respectively.

Competitors. We choose the following representative models as competitors: (1) general deep models: basic deep model (embedding layers + MLP), Wide & Deep (WD) [2], DeepFM [10]; (2) models with the attention mechanism or RNNs: DIN [45], DIEN [44], GRU4Rec [12], LSTM4Rec and MIMN [27]; (3) models with meta-learning methods: Meta-Embedding (Meta-E) [26], MeLU [18] and MAMO [5].

Metrics. We adopt logloss and AUC as evaluation metrics. The logloss is used to measure the convergence of the model which can be observed. For the AUC metric, it measures the probability of a positive sample score higher than a negative sample. In other words, AUC could measure the ranking ability of the model, which recommends the items with high click-through rates for users. For both metrics, the smaller the logloss value or the higher the AUC value, the better the model performance.

4.3 Results Analysis

For each dataset, we fine-tune the trained model iteratively by FT-A, FT-B and FT-C. In the meantime, we collect the test results after each iteration. The experimental result on MovieLens-1M dataset is shown in Table 2, while results on Book-Crossing and Avazu after fine-tuning with FT-C are show in Table 3 and Table 4, respectively. More details of results on Book-Crossing and Avazu datasets after fine-tuning with FT-A and FT-B refer to the Appendix. Besides, we record the results of all batches in the testing set, and plot an AUC distribution overall testing batches since AUC is a more important metric, which is shown in Figure 4.

4.3.1 ***Results on MovieLens-1M.*** As shown in Table 2, we can observe that our model achieves the best performance against other competitors over both WC and CW sub-datasets. Noteworthily, for the AUC metric, our model outperforms the second-best models

Table 3: AUC and loss on Book-Crossing dataset after fine-tuning with FT-C.

	WC		CW	
	loss	AUC	loss	AUC
Basic	4.7226	0.6169	4.6717	0.6223
WD	2.1817	0.6528	1.7849	0.6824
DeepFM	4.0464	0.6480	3.1006	0.6749
DIN	3.2609	0.6644	4.5779	0.6364
DIEN	2.1512	0.6799	2.6335	0.6511
GRU4Rec	2.3784	0.6690	3.6443	0.6588
LSTM4Rec	2.5298	0.6787	3.7772	0.6537
MIMN	3.7492	0.6306	0.6390	0.6264
Meta-E	1.8163	0.6331	2.7518	0.6444
MeLU	1.5202	0.6708	1.4536	0.6624
MAMO	2.7559	0.6254	2.1406	0.6585
Ours	0.4839	0.7185	0.5669	0.7123

Table 4: AUC and loss on Avazu dataset after fine-tuning with FT-C.

	WC		CW	
	loss	AUC	loss	AUC
Basic	1.3531	0.7035	0.2791	0.6953
WD	1.0081	0.7080	0.3409	0.6648
DeepFM	0.2960	0.7088	0.3044	0.6426
DIN	0.5405	0.7620	0.2837	0.7669
DIEN	0.4192	0.7809	0.4797	0.7478
GRU4Rec	0.7378	0.7078	0.3230	0.7457
LSTM4Rec	0.5691	0.7508	0.3096	0.7457
MIMN	0.2813	0.7121	0.2823	0.6887
Meta-E	0.7709	0.7289	0.3707	0.7394
MeLU	1.2893	0.6347	0.5960	0.7527
MAMO	1.2359	0.6112	1.6260	0.7143
Ours	0.0784	0.7938	0.1075	0.7969

after each fine-tuning iteration by 6.04%, 2.57%, 3.92%, respectively over WC, and by 0.51%, 3.90%, 5.83%, respectively over CW. Besides, for loss values, our model outperforms others after each fine-tuning iteration by 21.69%, 19.56%, 19.03%, respectively over WC, and by 26.18%, 30.04%, 31.20%, respectively over CW. These demonstrate the effectiveness of our proposed model and the meta-learning strategy.

For the WC sub-dataset, we could notice our model performs better after fine-tuning with FT-A than those after fine-tuning with FT-B or FT-C. The reason could be that our model holds the warm users' interests well during the training phase, so that only one fine-tuning is enough to predict on the test. On the contrary, for the CW sub-dataset, cold users have fewer interactions, and their interests or interests shifts are difficult to learn. Along with the fine-tuning, the performance of our model is improved and gains the best after FT-C. These demonstrate that the meta-learning strategy could make models learn to learn and adapt to new tasks fastly, even for those tough challenges.

For competitors, the attention-based method (DIN) and the RNN-based methods (GRU4Rec, LSTM4Rec, MIMN) outperform the naive deep models (Basic, WD and DeepFM) in general, which has been expected. The attention mechanism and RNN networks have proved their powerful ability in extracting item correlation and interests evolution, respectively. Both special structures are suitable for CTR tasks, which involve the sequence data and user interest shifts. It's worth noting that those attention-based or RNN-based methods obtain larger loss values than the naive deep models in both WC and CW sub-datasets. The reason could be that those models introduce more complicated modules, which probably result in overfitting. Relatively speaking, meta-learning could make the model less dependent on training tasks, and select parameters in a compromised manner. Therefore, the meta-learning methods (Meta-E, MeLU, MAMO) outperform the attention-based or RNN-based methods in some cases over both loss and AUC. However, the meta-learning methods gain a worse performance compared with ours. The good performance of ours may be because that the proposed meta-interaction module is well-suited to meta-learning and enhance each other to adapt to new sessions.

4.3.2 ***Results on Book-Crossing.*** According to Table 3, Figure 4(a) and Figure 4(b), our model outperforms other competitors over both WC and CW sub-datasets. Specifically, for the AUC metric, our model outperforms the second-best models after FT-C by 5.68% over WC, and by 4.38% over CW. Besides, for loss values, our model outperforms others after FT-C by 68.17% over WC, and by 11.28% over CW. Compared with the performance of the basic model on MovieLens-1M, we could observe that the basic model gains larger loss values on both WC and CW datasets. This means that the CTR tasks on Book-Crossing are more difficult than MovieLens-1M. In this case, all competitors get smaller loss values than the basic model, which demonstrates that attention mechanism, RNN networks or meta-learning strategy can all solve this problem to some extent. Especially, the meta-learning methods (Meta-E, MeLU, MAMO) obtain better improvement on the loss value. The reason could be that the meta-learning strategy makes the model own the ability to learn and not be trapped in some hard tasks. Besides, DIEN performs better than its performance on MovieLens-1M. The reason may be that its special combination of the attention mechanism and RNNs outstands when the user interactions turn to be sparser. Compared with it, we adopt another design perspective and construct our parallel composite structure in the meta-interaction module. With the meta-learning strategy, our model outperforms DIEN. This demonstrates that our framework is more suitable for the generalized cold-start problem.

4.3.3 ***Results on Avazu.*** According to Table 4, Figure 4(c) and Figure 4(d), our model performs better than other competitors over both WC and CW sub-datasets. For the AUC metric, our model outperforms the second-best models after FT-C by 1.65% over WC, and by 3.91% over CW. Besides, for loss values, our model outperforms others after FT-C by 72.13% over WC, and by 61.48% over CW. Since the user interactions in Avazu are sparser than MovieLens-1M and Book-Crossing, as we expected, DIEN performs well not only on Book-Crossing but also on Avazu. Besides, DIN gets the second-best

Table 5: Ablation study results after FT-C. "w/o Meta", "w/o Meta & LSTM" and "w/o Meta & LSTM" represent our model without the meta-learning strategy, the meta-learning strategy with the attention mechanism and the meta-learning strategy with LSTM, respectively.

Dataset	Model	FT-A				FT-B				FT-C			
		WC		CW		WC		CW		WC		CW	
		loss	AUC	loss	AUC	loss	AUC	loss	AUC	loss	AUC	loss	AUC
MovieLens-1M	Ours	0.4642	0.8316	0.3454	0.8054	0.4881	0.8013	0.3253	0.8326	0.4730	0.8167	0.3001	0.8480
	w/o Meta	0.5266	0.8016	0.4834	0.7876	0.5108	0.7819	0.3882	0.8217	0.5152	0.7831	0.4023	0.8205
	w/o Meta & Att	0.5336	0.8006	0.4203	0.8289	0.5103	0.7812	0.2679	0.9010	0.5144	0.7782	0.3481	0.8673
	w/o Meta & LSTM	0.5622	0.8110	0.5135	0.7817	0.5606	0.7931	0.4034	0.8101	0.5399	0.8027	0.4285	0.8190
Book-Crossing	Ours	0.4825	0.7193	0.5692	0.7176	0.4830	0.7193	0.5637	0.7212	0.4839	0.7185	0.5669	0.7123
	w/o Meta	0.7777	0.6848	0.6069	0.7133	0.7756	0.6842	0.6014	0.7172	0.7684	0.6839	0.6074	0.7050
	w/o Meta & Att	0.7508	0.6872	0.6043	0.7104	0.7486	0.6862	0.5930	0.7205	0.7397	0.6862	0.6040	0.7059
	w/o Meta & LSTM	0.9655	0.6443	0.6928	0.7008	0.9608	0.6463	0.6933	0.6990	0.9594	0.6458	0.6843	0.7012
Avazu	Ours	0.0790	0.7937	0.1082	0.7960	0.0788	0.7936	0.1075	0.8003	0.0784	0.7938	0.1075	0.7969
	w/o Meta	0.1068	0.7834	0.1900	0.7641	0.1007	0.7821	0.1881	0.7662	0.0987	0.7791	0.1890	0.7627
	w/o Meta & Att	0.1057	0.7799	0.1985	0.7676	0.0957	0.7779	0.1958	0.7693	0.0934	0.7750	0.1961	0.7660
	w/o Meta & LSTM	0.3826	0.7800	0.3128	0.7732	0.3592	0.7793	0.3087	0.7750	0.4075	0.7784	0.3050	0.7732

performance on CW. It may be because the attention mechanism is less affected by data sparsity than RNNs.

4.4 Ablation Study

To study the effectiveness of the meta-interaction module and the meta-learning strategy, we conduct ablation studies and design the competitors as follows:

- **w/o Meta**: Our framework is built without the meta-learning strategy.
- **w/o Meta & Att**: Our framework is built without the meta-learning strategy, and the meta-interaction module of our model discards the attention mechanism.
- **w/o Meta & LSTM**: Our framework is built without the meta-learning strategy, and the meta-interaction module of our model discards the LSTM layer.

Results are shown in Table 5, from which we could observe that our complete framework outperforms other competitors on the whole. Specifically, the improvements of our framework compared with the best competitor are listed as follows:

- **MovieLens-1M**: For WC sub-dataset, ours outperforms all other competitors. For CW sub-dataset, the competitor **w/o Meta & Att** seems to be the best on the AUC metric.
- **Book-Crossing**: For AUC metric, our model outperforms the second-best models after each fine-tuning by 4.67%, 4.82%, 4.71%, respectively over WC, and by 0.60%, 0.10%, 0.91%, respectively over CW. Besides, for loss values, our model gains the improvement by 35.74%, 35.48%, 34.58%, respectively over WC, and by 5.81%, 4.94%, 6.14%, respectively over CW.
- **Avazu**: For AUC metric, our model outperforms the second-best models after each fine-tuning iteration by 1.31%, 1.47%, 1.89%, respectively over WC, and by 2.95%, 3.26%, 3.07%, respectively over CW. For loss values, our model outperforms others by 25.26%, 17.66%, 16.06%, respectively over WC, and by 43.05%, 42.85%, 43.12%, respectively over CW.

We could see that the proposed meta-learning strategy plays an essential role in the proposed method. However, the meta-learning strategy should be equipped with a proper backbone, which can be seen from the worse performance of the meta-learning-based competitors. From the ablation study, we see that on different datasets, either LSTM with meta-learning or attention mechanism with meta-learning is important. However, our method provides a flexible way to combine both advantages and produce top performance in most cases.

5 CONCLUSION

In this paper, we raise a new setting, which is to solve the generalized cold-start problem for both unseen users and the new sessions from known users. In this case, we propose a recurrent meta-learning method. In our model, we build a parallel composite structure of the attention mechanism and RNNs layer to avoid the mutual propagation of errors and lead to error accumulation. Besides, a meta-learning strategy is adopted for session-based tasks. Compared with other mainstream deep models, our model consistently outperforms these models.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102000, in part by National Natural Science Foundation of China: U21B2038, 61931008, 61971016, 6212200758, 61976202, and 62006217, in part by the Fundamental Research Funds for the Central Universities, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB28000000, in part by China Postdoctoral Science Foundation: 2021T140653 and 2020M680651, and in part by mindspore[§], which is a new AI computing framework.

[§]https://www.mindspore.cn/

REFERENCES

[1] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *ACM International Conference on Web Search and Data Mining*. 108–116.

[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 *ACM Conference on Recommender Systems*. 7–10.

[3] Janghoon Choi, Junseok Kwon, and Kyoung Mu Lee. 2019. Deep meta learning for real-time target-aware visual tracking. In *IEEE/CVF International Conference on Computer Vision*. 911–920.

[4] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Workshop on Deep Learning*.

[5] Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming Zhu. 2020. MAMO: Memory-Augmented Meta-Optimization for Cold-start Recommendation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 688–697.

[6] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 515–524.

[7] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. In *International Joint Conference on Artificial Intelligence*. 2301–2307.

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

[9] L Franceschi, P Frasconi, S Salzo, R Grazzi, and M Pontil. 2018. Bilevel Programming for Hyperparameter Optimization and Meta-Learning. In *International Conference on Machine Learning*. 1568–1577.

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *International Joint Conference on Artificial Intelligence*. 1725–1731.

[11] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 355–364.

[12] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and D Tikk. 2016. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations*.

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

[14] Yuchin Juan, Damien Lefortier, and Olivier Chapelle. 2017. Field-aware Factorization Machines in a Real-world Online Advertising System. In *the World Wide Web Conference*. 680–688.

[15] Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic Meta-Embeddings for Improved Sentence Representations. In *Conference on Empirical Methods in Natural Language Processing*. 1466–1477.

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[17] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science* 350, 6266 (2015), 1332–1338.

[18] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1073–1082.

[19] Cheng-Te Li, Chia-Tai Hsu, and Man-Kwan Shan. 2018. A cross-domain recommendation mechanism for cold-start users based on partial least squares regression. *ACM Transactions on Intelligent Systems and Technology* 9, 6 (2018), 1–26.

[20] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *the World Wide Web Conference*. 661–670.

[21] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining explicit and implicit feature interactions for recommender systems. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1754–1763.

[22] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2018. DARTS: Differentiable Architecture Search. In *International Conference on Learning Representations*.

[23] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on Heterogeneous Information Networks for Cold-start Recommendation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1563–1573.

[24] Alex Nichol and John Schulman. 2018. On First-Order Meta-Learning Algorithms. *arXiv* abs/1803.02999 (2018).

[25] Feiyang Pan, Qingpeng Cai, Pingzhong Tang, Fuzhen Zhuang, and Qing He. 2019. Policy Gradients for Contextual Recommendations. In *the World Wide Web Conference*. 1421–1431.

[26] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm Up Cold-start Advertisements: Improving CTR Predictions via Learning to Learn ID Embeddings. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 695–704.

[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 *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2671–2679.

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

[29] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. 2018. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems* 37, 1 (2018), 1–35.

[30] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. 2019. Regularized evolution for image classifier architecture search. In *AAAI Conference on Artificial Intelligence*. 4780–4789.

[31] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, et al. 2019. Lifelong Sequential Modeling with Personalized Memorization for User Response Prediction. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 565–574.

[32] Steffen Rendle. 2010. Factorization machines. In *IEEE International Conference on Data Mining*. 995–1000.

[33] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 255–262.

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

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

[36] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A meta-learning perspective on cold-start recommendations for items. In *Neural Information Processing Systems*. 6904–6914.

[37] Ricardo Vilalta and Youssef Drissi. 2002. A perspective view and survey of meta-learning. *Artificial Intelligence Review* 18, 2 (2002), 77–95.

[38] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 968–977.

[39] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1–7.

[40] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: learning the weight of feature interactions via attention networks. In *International Joint Conference on Artificial Intelligence*. 3119–3125.

[41] Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Lifelong domain word embedding via meta-learning. In *International Joint Conference on Artificial Intelligence*. 4510–4516.

[42] Lina Yao, Quan Z Sheng, Xianzhi Wang, Wei Emma Zhang, and Yongrui Qin. 2018. Collaborative location recommendation by integrating multi-dimensional contextual information. *ACM Transactions on Internet Technology* 18, 3 (2018), 1–24.

[43] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *European Conference on Information Retrieval*. 45–57.

[44] 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 *AAAI Conference on Artificial Intelligence*. 5941–5948.

[45] 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 *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1059–1068.

[46] Barret Zoph and Quoc V Le. 2017. Neural architecture search with reinforcement learning. *International Conference on Learning Representations* (2017).