Revisiting Cold-Start Problem in CTR Prediction: Augmenting Embedding via GAN

Xuxin Zhang
Di Wang
Dehong Gao
Alibaba Group
Hangzhou, Zhejiang, China

Wen Jiang
Wei Ning
Yang Zhou
Alibaba Group
Hangzhou, Zhejiang, China

Chen Wang*
chenwang@hust.edu.cn
Huazhong University of Science and Technology
Wuhan, Hubei, China

ABSTRACT

Click-through rate (CTR) prediction is one of the core tasks in industrial applications such as online advertising and recommender systems. However, the performance of existing CTR models is hampered by the cold-start users who have very few historical behavior data, given that these models often rely on enough sequential behavior data to learn the embedding vectors. In this paper, we propose a novel framework dubbed GF2 to alleviate the cold-start problem in deep learning based CTR prediction. GF2 augments the embeddings of cold-start users after the embedding layer in the deep CTR model based on the Generative Adversarial Network (GAN) , and the obtained generator by GAN can be further fine-tuned locally to enhance the CTR prediction in cold-start settings. GF2 is general for deep CTR models that use embeddings to model the features of users, and it has already been deployed in real-world online display advertising system. Experimental results on two large-scale real-world datasets show that GF2 can significantly improve the prediction performance over three polular deep CTR models.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; Personalization; • **Computing methodologies** → *Generative and developmental approaches.*

KEYWORDS

Click-through rate prediction, cold-start problem, embedding, GAN.

ACM Reference Format:
Xuxin Zhang, Di Wang, Dehong Gao, Wen Jiang, Wei Ning, Yang Zhou, and Chen Wang. 2022. Revisiting Cold-Start Problem in CTR Prediction: Augmenting Embedding via GAN. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557684

*Corresponding author. This work was supported in part by the National Natural Science Foundation of China under Grant 61872416.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA.

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557684

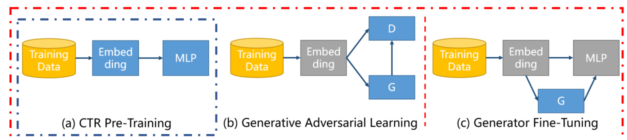

Figure 1: The pipeline of GF2. Blue dotted box shows the training of a regular deep CTR models while red dotted box shows the training of CTR models aided by GF2.

1 INTRODUCTION

Click-through rate (CTR) prediction, which aims to predict the likelihood that the recommended items will be clicked by a user, is an essential component in many online applications, such as e-commerce portals [5, 18] and social applications [15]. CTR prediction has made great progress in recent years [1, 3, 6, 9, 13, 17–19], benefiting from deep neural networks due to the strong expressive ability and the flexibility to learn rich representations from historical interactions. However, these deep CTR models suffer from the so-called cold-start problem [7], where some users have very few or even no historical behavior data to train a satisfactory model for them, yielding the personalized recommendation challenging.

Several methods have been proposed to deal with the cold-start problem in CTR, using the side information [8, 11] or the knowledge graph to automatically propagate users' potential preference [12]. Recently cross domain recommendation is utilized to improve the performance of CTR prediction in the target domain [4, 20], and meta-learning is adopted to transfer knowledge from other users/items to alleviate missing data of the target user [21]. However, existing solutions still suffer from the absence of practical and effective data augmenting to well address the cold start problem.

In this paper, we propose a general GAN-based Feature Generation Framework (GF2) to augment the embedding of cold-start users for deep learning based CTR models. Observing that most deep CTR models follow a similar structure of Embedding & Multi-Layer Perceptron (MLP) [18, 19] (c.f. Figure 1(a)), we propose to generate item embeddings, instead of item IDs in usual practice, after the embedding layer. We adopt the Generative Adversarial Network (GAN) to generate embeddings through a minimax game (c.f. Figure 1(b)), considering that GAN's compelling capability to generate realistic examples plausibly drawn from the existing distribution of real samples. Thereafter, we fine-tune the parameters of the obtained generator by GAN to ensure that the generated embedding could enhance the performance of specific CTR model

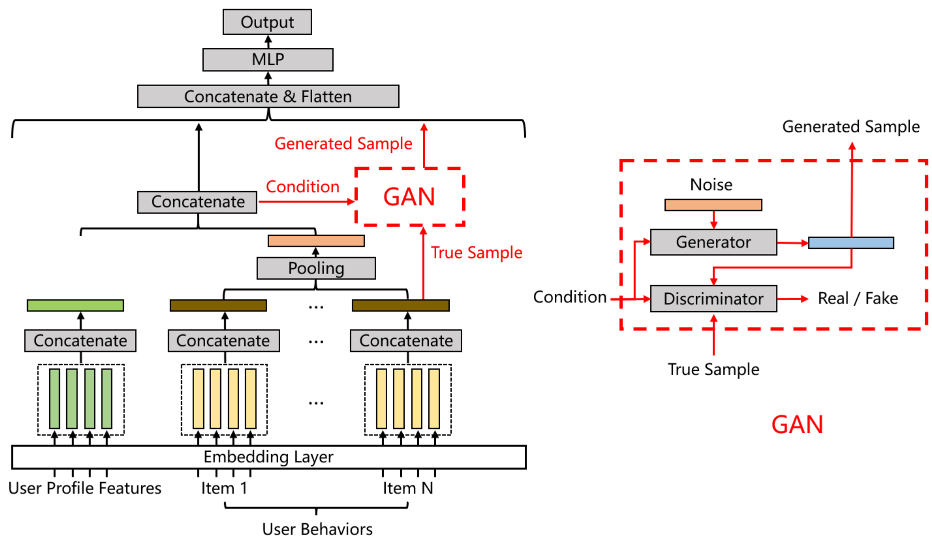

Figure 2: Illustration of model structure. Left part shows the structure of the base model (Embedding&MLP) aided by GF2. A GAN-based unit (the red dotted box) is introduced, which takes the concatenated embedding as the condition and outputs generated sample. Right part details the structure of GAN.

in cold-start setting (c.f. Figure 1(c)). By doing so, existing deep CTR models can be aided by GF2 to obtain a great performance gain on CTR prediction.

The main contributions of this paper are summarized as follows:

- We propose GF2, a general GAN-based feature generation framework to augment the embedding of cold-start users. Our framework is general for deep CTR models that use embeddings to model the features of users, and it has already been deployed in real-world online display advertising system of Alibaba Group.
- We propose to use GAN to generate cold-start users' sequence embeddings with side information as the condition. The obtained generator can be fine-tuned locally to enhance the prediction of specific CTR model in cold-start settings.
- We verify GF2 on two large-scale real-world datasets and release the code for reproduction[1]. Experimental results show that GF2 can significantly improve the prediction performance over three state-of-the-art deep CTR methods.

2 GF2 DESIGN

2.1 Embedding & MLP

Most of the popular deep models are similar to Embedding & MLP structures, in the domain of CTR prediction (c.f. the blue dotted part in Figure 1).

Embedding. Data is normally transformed into high-dimensional sparse binary features in the industrial CTR prediction online setting. With the advantage of embedding techniques, these original sparse features can be transformed into low dimensional dense continuous vectors, called embeddings. Similar to [19], we encode each feature into a single one-hot vector, which uses high-dimensional sparse binary encoding, based on its multi-group categorical form.

MLP. These one-hot vectors are transformed into a fixed-length real-valued dense vector by the pooling layer to adapt to the structure of MLP after the embeddings of the features are obtained. MLP is then used to automatically learn the feature combinations based on the given inputs that are concatenated with dense embeddings.

2.2 GF2 Implementation

Given a well trained CTR model (called base model) along with the well-trained embedding layers, GF2 mainly consists of two stages: (1) **Generative Adversarial Learning**: train a GAN to simulate the real distribution of behavior sequence embedding, and (2) **Generator Fine-Tuning**: fine-tune the generator to adapt to the CTR prediction task (c.f. Figure 1).

In ***Generative Adversarial Learning***, the generator aims to generate the plausible user historical behavior embedding, while the discriminator aims to discriminate the generated embedding and the real embedding (c.f. Figure 2). During the implementation, we build our framework based on Conditional GAN (CGAN), where the conditional inputs are concatenated with the embedding of user profile features and user historical behaviors. Formally, the training objective of Generative Adversarial Learning can be simulated as a minimax game:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)} \log\left(D(x|c)\right) + \mathbb{E}_{z\sim p_z(z)} \log\left(1 - D(G(z|c))\right) \tag{1}$$

where x represents the embedding of the user's last historical behavior, seen as the True Sample in Figure 2; c corresponds to the concatenated embedding as the condition, and z denotes a random noise vector, seen as the Condition and the Noise in Figure 2.

In this minimax game, the discriminator D aims to maximize the objective function whereas the Generator G aims to minimize it. Both G and D are represented by deep neural networks and trained by the stochastic gradient descent with minibatch and back-propagation algorithm. Mathematically, the objective function of

[1] https://www.dropbox.com/s/z4pf5ipugqc36cp/GF2-Code.zip?dl=0

the discriminator is denoted as:

$$J_D = -\mathbb{E}_{x \sim p_{data}(x)} \log(D(x|c)) - \mathbb{E}_{z \sim p_z(z)} \log(1 - D(G(z|c))) \quad (2)$$

and the objective function of the generator as:

$$J_G = \mathbb{E}_{z \sim p_z(z)} \log(1 - D(G(z|c))) \quad (3)$$

We train the model G and D alternately, keeping one fixed while updating the other. When the discriminator is unable to correctly discriminate the generated behavior sequence embeddings from the real ones, we completed the training of GAN. As a result of this stage, we expect that well-trained generator would capture the true distribution of real behavior sequence embedding.

In ***Generator Fine-Tuning***, the generator is trained on the CTR task by simply fine-tuning its parameters. Our goal is to ensure the generated embeddings could enhance the ability of the recommender, especially for cold-start users. The objective function of the generator in Generator Fine-Tuning is denoted as follows:

$$L_G = -\frac{1}{N} \sum_{i=1}^{N} y_i \log f_\theta(x_{gen}) - (1 - y_i) \log(1 - f_\theta(x_{gen})) \quad (4)$$

where x_{gen} is the enhanced input of MLP that adds the generated embeddings into the embedding of user historical behaviors when concatenated with user profile features, $y_i \in \{0, 1\}$ represents whether the user clicked the item and $f_\theta(x_{gen})$ is the prediction output of MLP, representing the prediction probability of the item to be clicked.

3 PERFORMANCE EVALUATION

3.1 Experiment Setup

3.1.1 Datasets. We conducted experiments on a public dataset as well as an industrial dataset collected from the online display advertising system of our company (see Table 1; L_{mean} means the mean length of the sequence).

Table 1: Dataset statistics.

Dataset	Users	Items	Instances	L_{mean}
Taobao	1,141,730	846,812	26,557,962	3.3
Industrial	5,167,854	32,539,198	349,609,464	1.9

Taobao Dataset[2] is constructed by user behavior logs from Taobao's recommender system. We take the click behaviors for each user and sort them according to time in an attempt to construct the behavior sequence. We use logs of the first 7 days as the training set (23,249,276 instances) and logs in the 8th day as the testing set (3,308,686 instances). We split the recent 100 user behaviors as user behavior sequences, as in [14, 18, 19].

Industrial Dataset is collected from the online display advertising system of Alibaba Group. We use logs of the first 30 days as the training set, while logs in the 31st as the testing set.

3.1.2 Base Models. We conduct experiments on the following mainstream CTR prediction models (also as base models):

- **DNN** is the basic deep learning model that follows the Embedding & MLP structure for CTR prediction.
- **DIN [19]**: applies attention to both the target item and user historical sequence to better model user interest subject to the target item.
- **DIEN [18]**: integrates GRU with attentional update gate for capturing the evolution trend of user interests and achieves state-of-the-art performance. It can be considered as an improved version of DIN.

3.1.3 Experiment Settings. We take the same implementations of the base models as that of DIEN [18] so that the results can be fairly compared. For all models, we take the Adam as the optimizer with the learning rate 0.001. Layers of fully connected network (FCN) and embedding layer are the same as in [18]. All experiments are repeated 5 times and average results are reported.

3.1.4 Metrics. We consider two common metrics, the AUC score (Area Under Receiver Operator Characteristic Curve), a widely used metric in CTR prediction [2, 18], which reflects the ranking ability of the model and is defined as follows:

$$AUC = -\frac{1}{|D^+| \times |D^-|} \sum_{x^+ \in D^+} \sum_{x^- \in D^-} (I(f_\theta(x^+) > f_\theta(x^-))) \quad (5)$$

where D_+ is the collection of all positive examples, and D_- is the collection of all negative examples. $f_\theta(\cdot)$ is the prediction result of the model and $I(\cdot)$ is the indicator function.

In addition, we follow [10, 16, 19] to adopt the RelaImpr metric to measure the relative improvement over the base model. For a random guesser, the value of AUC is 0.5, so RelaImpr is defined as:

$$RelaImpr = (\frac{AUC_{measured-model} - 0.5}{AUC_{base-model} - 0.5} - 1) \times 100\% \quad (6)$$

3.2 Performance of GF2

3.2.1 Universality of GF2. We first evaluate the performance of GF2 by testing on each base model before/after being aided by our GF2 on the Taobao dataset and industrial dataset. From the results in Figure 3, we find that GF2 can be applied to aid various base models well, even deploy in our online display advertising system with large scale industrial base model (Ind-BM). With different base models, GF2 is effective to improve the recommendation performance for cold-start users, showing satisfactory effectiveness and universality. Compared with the sequence models DIN and DIEN, GF2 has a higher performance gain on DNN. This observation indicates that the generated embeddings by GF2 indeed enhance the ability of the CTR prediction. We also notice that the gain is less obvious in DIEN, as there is no obvious sequential logic in our generated embeddings, and thus GF2 contributes less for base models that attach more importance to the sequence feature.

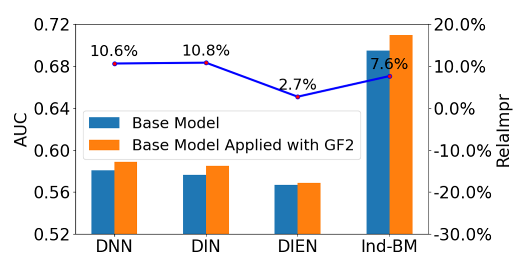

Figure 3: AUC of base model aided by GF2.

[2]https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

Table 2: AUC of cold start dataset aided by GF2.

Setting	DNN	DIN	DIEN
Original Dataset	0.5637	0.5632	0.5644
Cold Start Dataset	0.5594	0.5629	0.5571
Cold Start Dataset + GF2	0.5621	0.5636	0.5622

Table 3: AUC and execution time (hours) on Taobao dataset.

Model	Training Methods	AUC	Time
DNN	Base Model	0.5805	82.5
	Base Model with Repetition Training	0.5830	165.1
	Base Model Aided by GF2	**0.5890**	101.6
DIN	Base Model	0.5766	84.7
	Base Model with Repetition Training	0.5787	167.3
	Base Model Aided by GF2	**0.5849**	103.2
DIEN	Base Model	0.5671	83.3
	Base Model with Repetition Training	0.5683	168.6
	Base Model Aided by GF2	**0.5689**	103.0

3.2.2 *Cold Start Effect.*

To verify the effect of GF2 to the cold start problem, we next explore the influence of the cold start setting. We can see from Table 1 that the mean length of the sequence of Taobao dataset is 3.3, so we build the original dataset with users of length greater than 3.3, and a part of users are randomly selected to delete their sequence to construct the cold start dataset. In all three base models, the results of the cold start dataset aided by GF2 are better than without GF2, as can be observed in Table 2. From these results, we can see that GF2 enhances the embeddings of the cold-start users.

3.2.3 *Efficiency of GF2.*

From the above experiment results, it is shown that the application of GF2 indeed improves the effect of multiple base models. However, the framework also increases the parameters of model's network. From the point of view of efficiency, the necessity of GF2 has to be analyzed. Table 3 outlines the experimental results and execution time of different training methods based on multiple base models. It should be noted that the total parameters of all these three base models are approximately 144 million, and GF2 adds about 36 million parameters to the base models. So, the ratio of the trainable parameter for Pre-Training and other stages in GF2 is about 4:1. As a result, the framework aided by GF2 takes about a quarter of the additional time compared to repetition training and gets better results. The higher recommendation quality and less time consumed demonstrate the necessity of training a model with GF2 in cold-start scenarios.

3.3 Ablation Study

This section performs ablation studies to illustrate the effectiveness of the two stages including Generative Adversarial Learning (GAN) and Generator Fine-Tuning (Gen). Five training stages are designed for comparison as follows:

(1) CTR Pre-Training: training with only the deep CTR model (i.e., the base model).

(2) CTR Pre-Training + GAN: taking GAN to learn the real distribution of embedding after the deep model pre-training.

Table 4: AUC of different stages in GF2.

Model	DNN	DIN	DIEN
CTR Pre-Training	0.5805	0.5766	0.5671
CTR Pre-Training + GAN	0.5722	0.5751	0.5673
CTR Pre-Training + Gen	0.5880	**0.5854**	**0.5693**
CTR Pre-Training + Gen + GAN	0.5770	0.5744	0.5677
CTR Pre-Training + GAN + Gen	**0.5890**	0.5849	0.5689

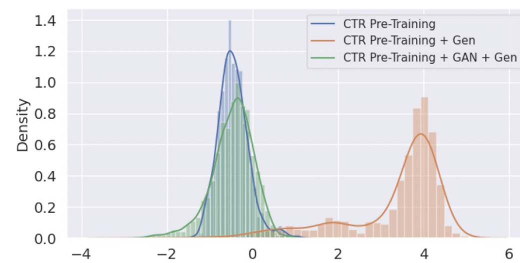

Figure 4: The kernel density estimate (KDE) of embedding of different stages of DNN aided by GF2.

(3) CTR Pre-Training + Gen: training with the generator based on the pre-training model.

(4) CTR Pre-Training + Gen + GAN: training the generator based on the pre-training model alone, then taking GAN to learn the real distribution of embedding.

(5) CTR Pre-Training + GAN + Gen: based on the trained deep CTR model, taking GAN to learn the distribution of real embedding and adjust the generator to adapt to the CTR task.

In Table 4, we show the ablation results for different training stages of multiple base models on the Taobao dataset. It is observed that the performance of base models becomes better after the Generator Fine-Tuning stage, while the addition of the Generative Adversarial Learning stage does not improve the prediction, which indicates that the Generator Fine-Tuning can significantly improve the recommendation performance for cold-start users.

In order to explore the role of the Generative Adversarial Learning stage, we analyzed the influence of this stage on the feature distribution of the generated sequence embeddings. We visually illustrate this problem by comparing the results of DNN with different training stages as shown in Figure 4. It can be seen that without the Generative Adversarial Learning stage, the sequence embeddings generated by the generator have significant differences from the real sequence embeddings in the feature distribution.

4 CONCLUSIONS

In this paper, we have proposed a novel feature generation framework (GF2) to address the cold-start problem in CTR prediction by augmenting embedding with GAN. The proposed GF2 is a general framework that can be applied to various deep CTR models that use embedding techniques. Experimental results on real-world datasets demonstrate that GF2 can significantly improve the prediction performance over three major deep CTR models.

REFERENCES

[1] 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 ACM DLRS*. 7–19.

[2] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. In *Proceedings of IJCAI*. 2301–2307.

[3] Wei Guo, Rong Su, Renhao Tan, Huifeng Guo, Yingxue Zhang, Zhirong Liu, Ruiming Tang, and Xiuqiang He. 2021. Dual Graph Enhanced Embedding Neural Network for CTR Prediction. In *Proceedings of ACM KDD*. 496–504.

[4] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of ACM CIKM*. 1563–1572.

[5] Feng Li, Zhenrui Chen, Pengjie Wang, Yi Ren, Di Zhang, and Xiaoyu Zhu. 2019. Graph intention network for click-through rate prediction in sponsored search. In *Proceedings of ACM SIGIR*. 961–964.

[6] Wentao Ouyang, Xiuwu Zhang, Li Li, Heng Zou, Xin Xing, Zhaojie Liu, and Yanlong Du. 2019. Deep spatio-temporal neural networks for click-through rate prediction. In *Proceedings of ACM KDD*. 2078–2086.

[7] Wentao Ouyang, Xiuwu Zhang, Shukui Ren, Li Li, Kun Zhang, Jinmei Luo, Zhaojie Liu, and Yanlong Du. 2021. Learning graph meta embeddings for cold-start ads in click-through rate prediction. In *Proceedings of ACM SIGIR*. 1157–1166.

[8] 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 *Proceedings of ACM SIGIR*. 695–704.

[9] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *Proceedings of IEEE ICDM*. 1149–1154.

[10] Shu-Ting Shi, Wenhao Zheng, Jun Tang, Qing-Guo Chen, Yao Hu, Jianke Zhu, and Ming Li. 2020. Deep time-stream framework for click-through rate prediction by tracking interest evolution. In *Proceedings of AAAI*. 5726–5733.

[11] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing Cold Start in Recommender Systems.. In *Proceedings of NeurIPS*. 4957–4966.

[12] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of ACM CIKM*. 417–426.

[13] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of International Workshop on Data Mining for Online Advertising*. 1–7.

[14] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep multi-interest network for click-through rate prediction. In *Proceedings of ACM CIKM*. 2265–2268.

[15] Ruobing Xie, Rui Wang, Shaoliang Zhang, Zhihong Yang, Feng Xia, and Leyu Lin. 2021. Real-time Relevant Recommendation Suggestion. In *Proceedings of ACM WSDM*. 112–120.

[16] Weinan Xu, Hengxu He, Minshi Tan, Yunming Li, Jun Lang, and Dongbai Guo. 2020. Deep interest with hierarchical attention network for click-through rate prediction. In *Proceedings of ACM SIGIR*. 1905–1908.

[17] Kai Zhang, Hao Qian, Qing Cui, Qi Liu, Longfei Li, Jun Zhou, Jianhui Ma, and Enhong Chen. 2021. Multi-interactive attention network for fine-grained feature learning in ctr prediction. In *Proceedings of ACM WSDM*. 984–992.

[18] 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 AAAI*. 5941–5948.

[19] 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 ACM KDD*. 1059–1068.

[20] Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021. Transfer-Meta Framework for Cross-domain Recommendation to Cold-Start Users. In *Proceedings of ACM SIGIR*. 1813–1817.

[21] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *Proceedings of ACM SIGIR*. 1167–1176.