DMBGN: Deep Multi-Behavior Graph Networks for Voucher Redemption Rate Prediction

Fengtong Xiao
fengtong.xiao@lazada.com
Alibaba Group
Singapore

Lin Li[*]
Weinan Xu[*]
boolean.ll@alibaba-inc.com
stella.xu@lazada.com
Alibaba Group, Singapore

Jingyu Zhao
jingyu.zhao@lazada.com
Alibaba Group
Singapore

Xiaofeng Yang
xiaofeng.yang@lazada.com
Alibaba Group, Singapore

Jun Lang
bill.lang@lazada.com
Alibaba Group, Singapore

Hao Wang
longran.wh@alibaba-inc.com
Alibaba Group, Singapore

ABSTRACT

In E-commerce, vouchers are important marketing tools to enhance users' engagement and boost sales and revenue. The likelihood that a user redeems a voucher is a key factor in voucher distribution decision. User-item Click-Through-Rate (CTR) models are often applied to predict the user-voucher redemption rate. However, the voucher scenario involves more complicated relations among users, items and vouchers. The users' historical behavior in a voucher collection activity reflects users' voucher usage patterns, which is nevertheless overlooked by the CTR-based solutions. In this paper, we propose a Deep Multi-behavior Graph Networks (DMBGN) to shed light on this field for the voucher redemption rate prediction. The complex structural user-voucher-item relationships are captured by a User-Behavior Voucher Graph (UVG). User behavior happening both before and after voucher collection is taken into consideration, and a high-level representation is extracted by Higher-order Graph Neural Networks. On top of a sequence of UVGs, an attention network is built which can help to learn users' long-term voucher redemption preference. Extensive experiments on three large-scale production datasets demonstrate the proposed DMBGN model is effective, with 10% to 16% relative AUC improvement over Deep Neural Networks (DNN), and 2% to 4% AUC improvement over Deep Interest Network (DIN). Source code and a sample dataset are made publicly available to facilitate future research[1].

CCS CONCEPTS

• **Information systems** → **Personalization**; **Recommender systems**; • **Applied computing** → *Electronic commerce.*

[*]Both authors contributed equally to this work.
[1]https://github.com/fengtong-xiao/DMBGN

KEYWORDS

Voucher; Redemption Rate Prediction; Deep Learning; Graph Neural Networks; Attention Mechanism; Multi-Behavior

1 INTRODUCTION

In E-commerce, vouchers have already become significant tools which not only drive sales, but also help user growth and enhances customer loyalty. An online voucher (or coupon in some literature) is a digital ticket that can be redeemed with discount or rebate when a product is purchased[2]. A typical voucher is characterized by two components: condition and discount. The former specifies requirements of a voucher to be applicable, e.g, the minimum number of items purchased or the minimum amount of money spent in an order, and the latter refers to the amount or percentage deductible when the voucher is applied. This work focuses on minimum-spend and discount-amount based vouchers.

Fig. 1 shows a typical user-voucher interaction scenario on an E-commerce platform: a user may do some online shopping before being assigned a voucher (i.e., pre-collection phase). After collecting the voucher (either manually or automatically), the user may continue to search, click or add-to-cart the items that he/she wants to buy with the voucher (i.e., post-collection phase). Finally, the user may redeem the voucher by purchasing some items (i.e., redemption phase), or simply leave the voucher expired without redemption.
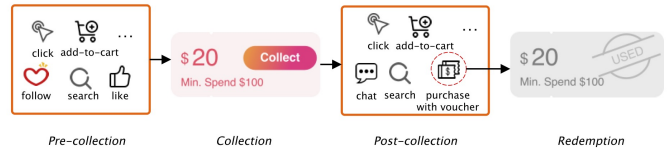

Figure 1: Illustration of interactions between users and vouchers in different phases.

[2]Retailers may limit the scope of a voucher to a specific pool of promoted items. This paper focuses on those voucher applicable to all the products in the platform.

The capability of predicting the user-voucher redemption rate (VRR henceforth in this paper), the likelihood that a user may redeem a voucher, would contribute to a successful voucher marketing in multiple ways:

- **Performance Forecasting**: Retailers are able to estimate the returns of a voucher (e.g. overall redemption rate, redeemed voucher count) before a campaign starts.
- **Budget Control**: Retailers are able to know when to stop distributing vouchers so that the marketing budget would not be over-utilized or under-utilized.
- **Personalized Distribution**: Retailers are able to distribute different voucher schema (e.g. condition and discount) to different customers to maximize the overall returns, by applying an optimization algorithm to the VRR results.

There are some preliminary studies [15, 20, 28] focusing on VRR prediction. Those studies solve voucher redemption rate prediction tasks by directly borrowing the ideas from user-item Click-Through-Rate (CTR) prediction models [6, 38]. A common practice is to treat a given voucher in the same way as treating an item, simply replacing item features by voucher features in model training and prediction.

Nonetheless, the VRR task is more challenging in several ways. Firstly, in item-wise CTR models, items in historical behavior and the item to be evaluated are homogeneous, while in VRR, the behavior involves heterogeneous relations between vouchers and items. Secondly, in CTR models, users' behavior on items has no obvious inter-dependent relationships. In voucher scenarios, by contrast, the user behavior is often triggered by the collected voucher and is strongly affected by the voucher condition and discount. As shown in Table 1, the behavior sequence in the post-collection phase becomes relatively longer than in pre-collection. Lastly, a user may have a multiple voucher redemption history, the long-term and short-term voucher usage preferences of the user need to be considered and balanced to better predict the VRR of the current voucher to be distributed (namely target voucher).

To tackle the above issues and shed light on the VRR prediction task, we propose a graph-based model Deep Multi-Behavior Graph Networks (DMBGN). To evaluate a user's VRR of the target voucher, DMBGN takes into consideration the user's historical interactions with vouchers. To model the structural user-voucher-item relationships, a User-Behavior Voucher Graph (UVG) is constructed for each user-voucher interaction activity. A UVG is a satellite-like heterogeneous graph containing nodes for both voucher and items in user behavior happening before and after the voucher collection. The techniques of Higher-order Graph Neural Networks are adopted to extract the high-level structural representation of a voucher activity, so that relationships like user-item preference, voucher-item dependency and user-voucher preference can be better distilled. To learn the user's long-term interests in voucher redemption, an attention network is applied on top of the historical UVG sequence. Finally, all the implicit representations of user, item, voucher and other side information are forwarded to a DNN network for the VRR prediction.

The main contributions of this paper are as follows:

- To our best knowledge, this is the first work to propose a model specialized for the VRR prediction task. The model, DMBGN, follows the user's mindset in a real voucher activity

Table 1: The average sequence lengths and the percentage difference of items from user behavior (click, add-to-cart and order) happening before and after voucher collection.

User Behavior	Avg. Sequence Length (before)	Avg. Sequence Length (after)	Avg. Sequence Length Diff % = (after)/(before)-1
Clicked Items	79.8	92.2	15.5%
Add-to-cart Items	10.9	13.7	25.2%
Ordered Items	1.8	3.6	105.7%

and represents the historical behavior happening both before and after voucher collection with a User-Behavior Voucher Graph (UVG).

- DMBGN is also the first work to utilize techniques of graph neural networks to learn the complex user-voucher-item relationships, and exploits an attention network to mine users' long-term voucher redemption preference with respect to the target voucher.
- Extensive experimental results conducted on three large-scale production datasets collected from E-commerce platform Lazada indicate promising performance of DMBGN. Compared with Deep Neural Networks (DNN), DMBGN achieves 10% to 16% relative AUC improvement. Compared with Deep Interest Network (DIN), DMBGN achieves 2% to 4% uplift in AUC as well.

2 RELATED WORK

2.1 Voucher-related Works

There are some preliminary studies in the literature addressing voucher-related problems. Those works can be divided into two types: i) the VRR prediction and ii) voucher distribution decision-making. For the first type, most of the studies adopt the CTR models widely utilized in item-wise recommendation. In [28], XGBoost model is used to solve the Online-to-Offline coupon redemption rate prediction, based on both user profile and coupon characteristic features. In [20], a sequence-based structure with attention-based mechanism is deployed, which considers long-term and short-term user behavior in addition to the voucher information. As discussed in introduction, our proposed model can better capture the more complex user-voucher-item relations with graph techniques, which is not presented in the previous works. For the second type, the task is to assign appropriate vouchers to different users based on the VRR prediction results and the total budget constraints. The most conventional optimization method for this task is the Multiple-Choice Knapsack Problem (MCKP) algorithm [18]. Some online learning algorithms can also be applied, such as online-MCKP [3], linear contextual bandits with knapsacks [1] and reinforcement learning based algorithms [9]. In this work, we focus on the first type of the problem on VRR prediction.

2.2 CTR and CVR Prediction Models

User-item Click-Through-Rate (CTR) prediction, which is to predict the probability of a user clicking an item, is the key task in the item-wise recommendation domain. The widely utilized models include Wide-and-Deep (WDL) [6] and Deep Factorization Machine (DFM) [13]. In recent years, attention-based structures to capture user's historical interest representation become a hot topic. Among them,

Deep Interest Network (DIN) [38], Deep Evolution Interest Network (DIEN) [37], Deep Hierarchical Attention Network (DHAN) [35], Deep Cross Networks (DCN) [32] and Behavior-sequence Transformer (BST) [5] are the representatives. There, the target and the attended objects are all items, and the behavior usually has no strong dependent relations. However, the VRR task needs to consider heterogeneous and inter-dependent relationships among users, items and vouchers.

Conversion rate (CVR), the possibility that a user will finally convert (e.g. purchase the item), is another important task in item-wise recommendation. The multi-task learning framework is a popular solution in this area. In [2, 22], MMoE and GMCM models utilize the shared bottom structure and task-specific upper structures to optimize CTR and CVR together. In [33], ESMM2 follows a similar strategy but instead optimizes CTR and Click-Through-Conversion Rate (CTCVR). Although multi-task learning can help to estimate the conversion before the target is exposed, it is not appropriate for the VRR task. Users usually have no objection to collect an assigned voucher, and under some circumstances, the voucher is auto-collected according to the platform's strategies. Therefore, it is difficult or even impossible to co-train the collection rate and the redemption rate in the multi-task manner.

There are also some works which take multiple user behaviors into account for more accurate CTR and CVR prediction. In these works, various behaviors such as clicks, favorites, add-to-cart, purchases, etc are considered and specific techniques are applied to model user-item relationships for better inference, such as hetergeneous graph convolution network [2, 16], cascading ordinal behavior structure [4, 10] and transformer architecture [34]. However, our work for VRR aims to model user-voucher-item relationships through a graph network, which focuses more on capturing different user behavior patterns happening before and after voucher collection.

2.3 Graph Embedding and Related Models

Recent years have seen the surge of research interests in deep graph-network techniques. A series of graph embedding techniques such as DeepWalk [26], Node2Vec[12] and MetaPath2Vec [7] have been widely utilized in various applications. More recently, convolutional network structures become popular which can dig deeper structural relationships from the graph. A number of models such as WL GNN [24], GraphSage [14], Graph-Attention Networks (GAT) [31], Graph Convolutional Networks (GCN) [19] and Gated Graph Neural Networks (GGNN) [21] have been successfully deployed in the fields of social networking, advertising, risk assessment and recommendation systems. For the CTR tasks, DHGAT[25] employs a heterogeneous graph structure to model the relationships among users, queries, shops and items. ATBRG [8] utilizes graph pruning techniques to get more effective recommendations. GMCM [2] adopts GCN to generate the hidden representations of various types of behavior such as click, review and add-to-cart.

3 METHODOLOGY

This section introduces the DMBGN model for VRR prediction. It starts with the problem definition, followed by the illustration of User-Behavior Voucher Graph (UVG) and the overview of the model architecture. Next, details about the graph neural networks on UVG
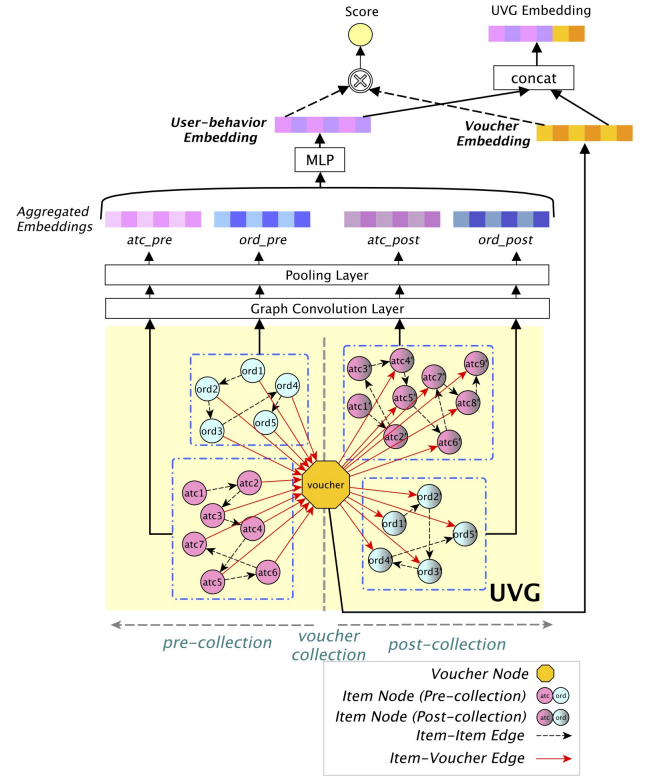

Figure 2: Illustration of User-Behavior Voucher Graph (UVG). The voucher node in the center represents the collected voucher. Item nodes from user behavior are subgrouped by action types (*atc* and *ord*) and phases (pre-collection and post-collection).

are given. Then, the attention mechanism which aggregates the UVG sequences is explained. Finally, the loss function used for model training is presented.

3.1 Problem Setup

Given a set $< \mathcal{U}, O, \mathcal{I}, C >$, $\mathcal{U} = \{u_1, u_2, u_3, \cdots, u_m\}$ denotes the set of m users, $O = \{o_1, o_2, o_3, \cdots, o_n\}$ denotes the set of n vouchers, $\mathcal{I} = \{i_1, i_2, i_3, \cdots, i_t\}$ denotes the set of t items, $C^{u,o} = \{c_1^{u,o}, c_2^{u,o}, c_3^{u,o}, \cdots, c_k^{u,o}\}$ denotes k types of behavior happening before and after user u collects voucher o. A behavior is an item from $\mathcal{I}$, associated with an action type (e.g, click, add-to-cart, order). Further, $C^{u,o}$ is divided into two subsets $C_b^{u,o}$ and $C_a^{u,o}$, representing the behavior happening before (*b*) and after (*a*) voucher collection, respectively. In $C_a^{u,o}$, all the behaviors after the voucher collection and before the voucher redemption (or expiration) are included. In $C_b^{u,o}$, the nearest behavior items before voucher collection, up to a specified number, are included. A voucher session describes a voucher collected by the user and the associated behavior, which can be defined as a tuple of $S(u, o, C^{u,o})$.

The problem of VRR predication is to calculate the probability of a user to redeem a voucher, given the behaviors before user collects the voucher. Formally, the voucher redemption rate is defined as $P(y_{RR} = 1 | u \in \mathcal{U}, o \in O, C_b^{u,o})$, where $y_{RR} = 1$ if and only if user u redeems the voucher o.
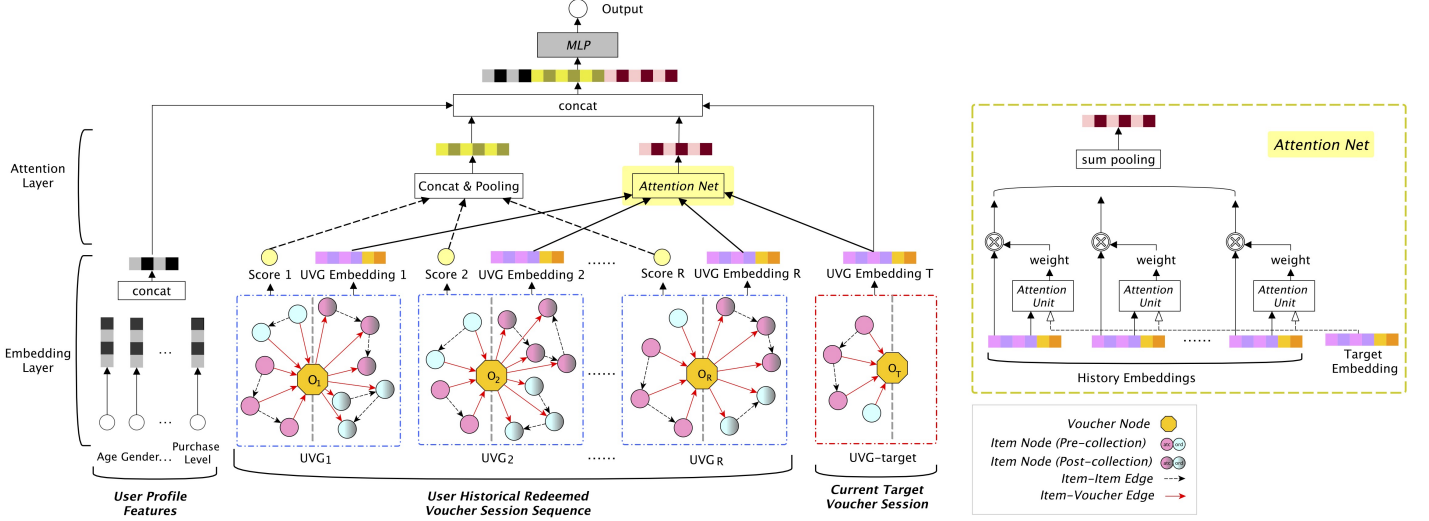
Figure 3: The architecture of DMBGN. The input consists of user profile features, UVG sequences representing user historical redeemed voucher sessions, and target voucher session. For each UVG, an embedding and its score are obtained. An attention layer is built on top of the UVG sequences to obtain the aggregated embeddings. The outputs from the three parts are concatenated and forwarded to the MLP for the VRR prediction of the target voucher O_T.

3.2 User-Behavior Voucher Graph (UVG)

To model the user-voucher-item relationships, a User-Behavior Voucher Graph (UVG) is constructed for each voucher session $S(u, o, C^{u,o})$. The graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ represents the node set and $\mathcal{E}$ represents the edge set. $\mathcal{V}$ contains a node for the voucher, denoted as $\mathcal{V}_o$ and the nodes for items in $C^{u,o}$, denoted as $\mathcal{V}_I$. Item nodes $\mathcal{V}_I$ can be further distinguished according to action timestamps as the pre-collection (b) and post-collection (a) phases, and according to the action types, i.e., add-to-cart (atc) and order (ord) [3]. Thus, $\mathcal{V}_I$ is divided into four sub-groups $\mathcal{V}_{I,atc}^b$, $\mathcal{V}_{I,ord}^b$, $\mathcal{V}_{I,atc}^a$ and $\mathcal{V}_{I,ord}^a$ along the two dimensions.

Edges in $\mathcal{E}$ have two types. The first type connects the item nodes within the above four sub-groups of $\mathcal{V}_I$, while the other type connects the item nodes to the voucher node. As shown in Fig. 2, a UVG is a satellite-like structure where the center is the voucher node surrounded by four zones. These four zones represent four types of relations, corresponding to the four sub-groups of $\mathcal{V}_I$, respectively. The central voucher node is only connected to the closest (in event timestamp) Z item nodes in every zone, while in each particular zone, an item node is connected to its closest node. UVG is a directed graph, and the direction of each edge follows the corresponding chronological order.

3.3 Model Architecture

The overall framework of DMBGN is presented in Fig. 3. The model takes user's profile information, historical voucher sessions information, and the current voucher session information as inputs. The left bottom part is for the user profiles features handled by an embedding layer. The central bottom is for the user's historical redeemed voucher sessions, while the right bottom is for the

---

³We also conducted experiments considering nodes related to *click* action type, but the results do not show any further improvement. This may attribute to the noises in the large amount of clicked items, and the weak correlations between clicked items and collected vouchers.

---

session of the target voucher to be predicted. The information and relationships about the voucher and its surrounding behavior are represented by UVG. Graph Neural Networks (GNN) are applied on the UVG to generate the embedding, as well as a score reflecting the quality of the embedding. On top of a user's historical UVG sequence, an attention network is introduced to obtain the aggregated embedding, which represents the user's voucher usage interest with respect to the target voucher. An addition output of the UVG sequence is also created by concatenating and pooling all the UVG scores in the sequence.

The outputs of the user profiles, historical voucher sessions and target voucher session are concatenated together and forwarded to an MLP layer for final redemption rate prediction. The result is a 0-to-1 value indicating the probability that a user would redeem a voucher. The parameters of DMBGN are learned on labelled training samples and can be applied for the later VRR prediction.

3.4 Graph Embedding for UVG

3.4.1 *Higher-order Graph Neural Networks.* When each UVG is constructed, graph neural networks are applied to extract the high-level representations. Higher-order Graph Neural Networks with Weisfeiler-Leman Algorithm [24] have the advantage of fast computation, easy tuning and robustness, and have been widely used in production pipelines. Hence Higher-order GNN is adopted here to conduct the convolution operations which update each node $v \in \mathcal{V}$ in UVG as follows:

$$f^{(l)}(v) = \sigma\big(f^{(l-1)}(v) \cdot W_1^{(l)} + \sum_{w \in N(v)} f^{(l-1)}(w) \cdot W_2^{(l)}\big) \quad (1)$$

where l represents l^{th} layer, and $N(v)$ represents the neighbors of v. $W_1^{(l)}$ and $W_2^{(l)}$ represent the GNN weight parameters for l^{th} convolution layer. $\sigma(\cdot)$ represents the sigmoid activation function.

The first layer node embedding $f^{(0)}(v)$ is obtained by separate embedding layers for the input item and voucher features. For an

item, the input features include item id, category id and item price level; for a voucher, the input features include the corresponding marketing activity id and schema values (i.e., minimum-spend and discount-amount).

The node embeddings generated by Equation (1) are further aggregated into the user-behavior embedding $e_{UVG,b}$, the voucher embedding $e_{UVG,o}$, and the final UVG embedding e_{UVG}. Recall the item nodes in UVG can be divided into four sub-groups. We calculate the embedding for a sub-group g as follows:

$$e_g = e_{g,avg} \mathbin{||} e_{g,max} \tag{2}$$

where $e_{g,avg}$ and $e_{g,max}$ represent the average and the maximization of all the node embeddings in the sub-group g. These two results are concatenated to form the sub-group's final embedding e_g. Next, the embeddings of four sub-groups, denoted as atc_pre, ord_pre, atc_post and ord_post are concatenated and forwarded to an MLP to derive $e_{UVG,b}$, that is:

$$e_{UVG,b} = MLP(atc\_pre \mathbin{||} ord\_pre \mathbin{||} atc\_post \mathbin{||} ord\_post) \tag{3}$$

The voucher embedding $e_{UVG,o}$ is simply the embedding from the voucher node. $e_{UVG,b}$ and $e_{UVG,o}$ are concatenated to obtain the final UVG embedding e_{UVG} as in Equation (4). In addition, a score for a historical UVG is generated from the dot-product between $e_{UVG,b}$ and $e_{UVG,o}$ as in Equation (5). The score, namely s_{UVG}, represents how well the voucher embedding alone can reflect the related behavior associated with this voucher.

$$e_{UVG} = e_{UVG,b} \mathbin{||} e_{UVG,o} \tag{4}$$

$$s_{UVG} = \sigma(e_{UVG,b} \cdot e_{UVG,o}) \tag{5}$$

where the $\sigma(\cdot)$ represents the sigmoid function.

The first-layer node embeddings $f^{(0)}(v)$ for items and vouchers can either be learned from scratch with the main task, or be trained independently in advance. In this work, we pre-train these initial item node embeddings and voucher embeddings separately.

3.4.2 *Pre-training of Item Embedding*.
Following the previous works in [26, 27, 30], a meta-path type graph embedding algorithm is applied to generate the item embeddings. In order to avoid overfitting, the algorithm is run on all the logged user behavior data from our platform. To enrich the expression of users' item sequences, side-information features such as $category\_id$, $brand\_id$, and $shop\_id$ are added for training. The pre-training is conducted separately for add-to-cart and order behavior in this paper, so that the same item of different behavior types has different embeddings. The learned embeddings are used to initialize the item nodes in both historical and target UVGs.

3.4.3 *Pre-training of Voucher Embedding*.
The historical UVGs in DMBGN represent the voucher-redeemed sessions. However, there may exist an overwhelming number of non-redeemed voucher sessions in the voucher usage data set. In this work, we utilize all the redeemed and non-redeemed voucher sessions from the platform to pre-train the voucher embeddings. A learned voucher embedding is expected to be capable of representing its related usage behavior, so it should be close to its corresponding behavior embedding in a semantic space if the voucher is redeemed, and far away otherwise.

For each user voucher session, a UVG is constructed as described in Section 3.2. The item embeddings input to UVG are implemented as in Section 3.4.2, while the voucher embedding is randomly initialized. Recall that s_{UVG}, the score of the UVG represents the similarity between $e_{UVG,b}$ and $e_{UVG,o}$. Loss function for voucher embedding training is defined as:

$$L_{pre-voucher} = \frac{1}{N} \sum_{n=1}^{N} l(y_{UVG}^n, s_{UVG}^n) \tag{6}$$

where N represents the total size of the UVG pre-trained samples, and n represents the nth sample. y_{UVG} is a 1-or-0 label of UVG indicating the ground-truth whether the user redeems this voucher or not. $l(\cdot)$ represents the binary cross-entropy loss function.

After training, the initial embeddings of the voucher nodes are learned. Note that the above pre-training involves the ground-truth label whether a user redeems a voucher. Therefore, it is only conducted in the historical UVGs, but not in the target UVG. For all the target UVGs, the voucher embeddings are randomly initialized and learned from scratch with the main task.

In addition to the voucher embeddings, the weight parameters of the Higher-order Graph Neural Networks are learned during the pre-training of UVGs. These weights are used for DMBGN network initialization and to be further fine-tuned with the main task.

3.5 Attention Networks on UVG sequences

A user may have multiple previous voucher redemption sessions and thus may have a sequence of historical UVGs. To discover the user's long-term voucher usage preference (e.g. "What kind of voucher the user will be interested in?" and "What kinds of items will the user purchases with the voucher?"), an attention layer is added on top of the historical and the target UVGs. The *AttentionNet* proposed in the work of DIN [38] is adopted as follows:

$$\begin{aligned} w^r &= AttentionUnit(e_{UVG}^r, e_{UVG}^T, \phi), r = 1, 2, \cdots, R \\ h_T &= \sum\nolimits_{r=1}^{R} w^r \cdot e_{UVG}^r \end{aligned} \tag{7}$$

here, e_{UVG}^r and e_{UVG}^T are the embeddings of the rth historical UVG and the target UVG, respectively, and h_T is the high-level representation of historical UVG sequences with respect to the target voucher O_T. The *AttentionUnit* in Fig. 3 is a feed-forward neural network with scalar output w^r as the attention weight, and the network parameters are represented as ϕ. Apart from the two input embedding vectors, *AttentionUnit* adds their outer-product and difference as well, which helps model the interactions between an input pair as mentioned in the original DIN model.

The learned attention network gives higher weights to the historical voucher sessions that are semantically closer to the target session. These historical sessions include the complete behavior information, and thus could better reflect the user's long-term voucher usage preference with respect to the target voucher. Referring to such sessions gives valuable clues whether a voucher will be redeemed in the end.

3.6 Loss Function

To train the DMBGN model, two loss functions are considered. The first one, L_{target}, evaluates the loss based on the difference between the predicted voucher redemption result $\hat{y}_{RR}^n$ and the ground-truth

label y_{RR}^n, where n represents the nth training sample. The loss function is formulated as:

$$L_{target} = \frac{1}{N} \sum_{n=1}^{N} -(y_{RR}^n \log \hat{y}_{RR}^n + (1 - y_{RR}^n) \log(1 - \hat{y}_{RR}^n)) \quad (8)$$

where N is the total number of training samples.

Another loss function is introduced for regularization. Recall in DMBGN, each historical UVG session outputs a score s_{UVG} indicating whether the voucher and the behavior embeddings are semantically close. It is expected that the voucher embedding can reflect the related behavior, the score should be as close to 1 as possible because each historical UVG corresponds to a redeemed session. Inspired by the work in [37], for better supervision of the UVG graph network training during the main task, the auxiliary loss function is defined as:

$$L_{aux} = \frac{1}{N} \sum_{n=1}^{N} \left(\frac{1}{R_n} \sum_{r=1}^{R_n} -log(s_{UVG}^r) \right) \quad (9)$$

where R_n denotes the history UVG sequence length from the nth sample, and r represents the rth UVG in a user's history UVG sequence. Therefore, the final loss L is the weighted sum of L_{target} and L_{aux}, represented as follows:

$$L = L_{target} + \alpha \cdot L_{aux} \quad (10)$$

where α is the trade-off coefficient.

4 EXPERIMENTS

4.1 Dataset

For the VRR prediction task, there is no public dataset so far. In this work, we use the large-scale dataset from the real voucher log data from Lazada, a leading South-East Asia (SEA) E-commerce platform of Alibaba Group. The data covers Lazada's voucher marketing activities during campaign periods such as Double 11 and Double 12 from 2019 to 2020. We select the data from three countries in SEA, named as Region-A, Region-B and Region-C, respectively. The three datasets contain users' platform voucher collection and redemption logs as well as users' behavior occurring both before and after voucher collection. To preserve adequate amount of user behavior without introducing too much noise, the 95% percentile of users' behavior length is retained. For each user, at most 45 add-to-cart and 20 order items before or after voucher collection are reserved. The three datasets are of different sizes, and reflect various user-voucher engagement degrees. Overall, 44.2% (Region-A), 11.6% (Region-B) and 31.2% (Region-C) of the samples have at least one previous voucher redemption record.

Each sample in the dataset is represented as a unique <user_id, voucher_id> pair, with a label 1 or 0 indicating if the user redeemed the voucher or not. A voucher_id represents an individual voucher marketing activity. Each sample is associated with the user profile features (predicted age level, predicted gender, predicted purchase level, and user's order statistics), voucher features (minimum-spend, discount-amount, collection and redemption timestamps) and item features (pre-trained embeddings, brand_id, category_id, price level). All numerical features used in the following experiments are normalized through z-normalization and all sparse id features are encoded using LabelEncoder.

Table 2: Statistics of three datasets from different regions.

Dataset	Sessions	Users	Items	Vouchers	Training Samples	Testing Samples
Region-A	3.4M	730K	7.2M	1k	2.7M	687K
Region-B	14M	6.5M	19M	0.5k	11M	2.8M
Region-C	24M	7.0M	15.5M	1k	19.4M	4.9M

We have published a desensitized subset of original dataset Region-C for public use. Table 2 summarizes the statistics of three datasets, a full description of the dataset is also available on GitHub.

4.2 Compared Models

To evaluate the performance of proposed method, we compare DMBGN with the following models which are commonly adopted in item CTR prediction tasks.

- **LR:** Logistic Regression [23] is a shallow model. Only dense features from user profiles and voucher information are used in the model.
- **GBDT:** Gradient Boosting Decision Tree [17] is used to assess the performance of non deep-learning algorithms. Similarly, only dense features are used.
- **DNN:** The Deep Neural Network is used as the first baseline taking both dense features and embedding of sparse id features into the model.
- **WDL:** Wide and Deep model [6] is widely accepted in real industrial applications. Compared with DNN, it has an additional linear model besides the deep model. All dense and sparse features are used in both wide and deep side.
- **DIN:** Deep Interest Network [38] is an attention-based model in recommendation systems that has been proven successful in Alibaba. We use this as our second baseline, replacing the user's historical item sequences with user's historical voucher sequences to adapt to the VRR prediction task. All dense and sparse features are used in the model along with the attention part.

4.3 Evaluation Metrics

Two widely used metrics for binary classification tasks are adopted in our experiments: AUC and Logloss. Area under receiver operator curve (AUC) measures the probability of ranking a random positive sample higher than a negative one. Logloss is also utilized which reflects the prediction inaccuracy of voucher redemption rate. Each experiment is repeated 5 times. Mean and standard deviation of AUC and mean of Logloss are reported.

In addition, to measure the relative AUC improvement over models, RelaImpr [36, 38] is also used which is defined as:

$$RelaImpr = \left(\frac{AUC(measured\ model) - 0.5}{AUC(base\ model) - 0.5} - 1 \right) \times 100\% \quad (11)$$

4.4 Hyper-parameter Settings

All experiments in this paper use Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and learning rate $lr = 0.001$. 8 GPUs are used in the experiment with batch size of 1024. In the *MLP* layer across different deep models, the hidden units are (128, 64); dropout rate is set as 0.5; $L2$ regularization is applied to avoid over-fitting with

Table 3: Model performance evaluated by AUC on three production datasets. The Relative Improvement (RelaImpr) takes DNN and DIN as baseline models. Best results of all methods are highlighted in boldface.

Model	Region-A				Region-B				Region-C			
	AUC mean ± std	RelaImpr (DNN)	RelaImpr (DIN)	Logloss mean	AUC mean ± std	RelaImpr (DNN)	RelaImpr (DIN)	Logloss mean	AUC mean ± std	RelaImpr (DNN)	RelaImpr (DIN)	Logloss mean
LR	0.7491 ± 0.00005	-9.04%	-14.43%	0.4385	0.7805 ± 0.00009	-10.74%	-17.93%	0.3712	0.7565 ± 0.00011	-16.66%	-26.39%	0.3202
GBDT	0.7538 ± 0.00004	-7.34%	-12.83%	0.4215	0.7924 ± 0.00004	-6.94%	-14.44%	0.3610	0.7982 ± 0.00005	-3.12%	-14.43%	0.2933
DNN	0.7739 ± 0.00000	0.00%	-5.93%	0.4156	0.8142 ± 0.00000	0.00%	-8.06%	0.3478	0.8078 ± 0.00000	0.00%	-11.67%	0.2966
WDL	0.7752 ± 0.00004	0.47%	-5.49%	0.4145	0.8188 ± 0.00004	1.46%	-6.72%	0.3452	0.8110 ± 0.00009	1.05%	-10.74%	0.2948
DIN	0.7912 ± 0.00029	6.30%	0.00%	0.4021	0.8417 ± 0.00017	8.77%	0.00%	0.3248	0.8485 ± 0.00022	13.22%	0.00%	0.2692
DMBGN-AvgPooling	0.7961 ± 0.00017	8.09%	1.68%	0.3975	0.8430 ± 0.00015	9.15%	0.36%	0.3238	0.8497 ± 0.00026	13.60%	0.34%	0.2681
DMBGN-Pretrained	0.8012 ± 0.00014	9.97%	3.45%	0.3954	0.8444 ± 0.00011	9.62%	0.78%	0.3217	0.8530 ± 0.00023	14.67%	1.29%	0.2658
DMBGN	**0.8034 ± 0.00023**	**10.76%**	**4.20%**	**0.3921**	**0.8486 ± 0.00021**	**10.94%**	**2.00%**	**0.3184**	**0.8591 ± 0.00015**	**16.65%**	**3.04%**	**0.2614**

coefficient as 0.1. In *Attention Net*, PReLU [38] is selected as the activation function with one layer of 64 hidden units. The embedding dimension of sparse id features is 16. The Graph Neural Networks for UVGs adopt a graph convolution layer and a top-K pooling layer [11] with ratio = 0.9. The loss trade-off coefficient is set as $\alpha = 1.0$.

Besides, as 99.10% of users have not more than 6 historical redeemed vouchers, the maximal length of historical UVG sequence is set as $R = 6$. The number of nodes connected to the voucher node in UVG mentioned in Section 3.2 is set as $Z = 6$.

5 RESULTS AND DISCUSSIONS

In this section, we analyze the overall performance of DMBGN and other compared models. The following four questions are also proposed to be answered:

- **Q1:** Are graph neural networks in DMBGN helpful in improving the overall performance?
- **Q2:** How does the user behavior happening before and after voucher collection help improve the overall performance of DMBGN?
- **Q3:** How does user activeness in terms of historical UVG sequence length affect the DMBGN performance?
- **Q4:** How are the user-voucher-item structural relationships distilled by Higher-order Graph Neural Networks?

The experimental results of DMBGN and the other compared models are summarized in Table 3. Besides, we also list the results of two variants of DMBGN, which use the same network structure but different embedding generation methods:

- **AvgPooling**: Instead of using Higher-order Graph Neural Networks to model user-voucher-item relationships, it directly takes an average of pre-trained item embeddings from user behavior happening both before and after voucher collection. For target UVG, it only takes an average of pre-collection item embeddings.
- **Pretrained**: It uses the same weight parameters of Higher-order GNN learned during the voucher embedding pre-training as mentioned in Section 3.3. But the values of weight parameters are not further updated during the main task training.
- **DMBGN**: The proposed model in this work, which loads the pre-trained GNN network including the item and voucher node embeddings. The GNN network parameters are further fine-tuned according to the final training loss.

5.1 Overall Performance

Based on the results in Table 3, we have the following observations:

- DMBGN consistently outperforms other baseline models across all three datasets in terms of AUC and Logloss, the standard deviation of AUC is smaller than 0.001 indicating reliable results. Especially, DMBGN achieves over 10% to 16% relative improvement in AUC compared to the most commonly used DNN model. It supports that DMBGN is capable of capturing the complex user-voucher-item relationship with User-Behavior Voucher Graph (UVG) well, and such relationship is commonly ignored in other models.
- Compared with the competitive model DIN, DMBGN achieves around 2% to 4% relative uplift in AUC. Recall that in the VRR tasks, DMBGN does not simply replace items by vouchers, but constructs the UVG and utilizes graph neural networks to better model the pattern between user behaviors and voucher collection.
- We observe that the AUC of DMBGN improves from 0.8034 to 0.8591 as the training size grows from 3.4 million to 24 million, indicating that DMBGN is capable of handling large-scale dataset, and performs better with more training samples involved. Across three datasets, the AUC relative improvement against DIN is larger in regions that are more active in voucher usage, specifically, 4.20% uplift in Region-A, compared to 2.00% in Region-B and 3.04% in Region-C.

5.2 Ablation Study (Q1)

To further demonstrate the effectiveness by introducing the graph structure, we compare the performance of DMBGN with its two variants. The results are shown in the last three rows in Table 3:

- Taking *AvgPooling* as baseline, *Pre-trained* achieves the relative AUC improvement of 1.7%, 0.4%, 0.9% as calculated in Equation (11) on dataset Region-A, B, C respectively. Recall that *AvgPooling* and *Pre-trained* take different strategies to generate the user behavior embedding $e_{UVG,b}$, by simple mean pooling and by graph techniques, respectively. The out-performance of *Pre-trained* over *AvgPooling* supports that the construction of UVG and Higher-order GNN could better capture user-voucher-item relationships.
- After fine-tuning of Higher-order GNN network parameters, DMBGN further improves the AUC over *Pre-trained* version by 0.7%, 1.2%, 1.7% on dataset Region-A, B, C, respectively. This improvement indicates that the graph network can adaptively learn from the final loss.

5.3 Post-collection User Behavior Study (Q2)

Unlike item-wise recommendation, the user behavior related to voucher collection plays an important role in final redemption decisions. Hence, we conduct experiment to compare the influence from behavior in different phases (i.e., pre-collection and post-collection) on the overall performance of DMBGN. The results are summarized in Table 4.

Table 4: Experiment results of DMBGN for including and excluding post-collection user behavior on three datasets.

Model	Region-A		Region-B		Region-C	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
DMBGN w/ pre-collection behavior only	0.8014	0.3937	0.8479	0.3191	0.8557	0.2639
DMBGN w/ pre- and post-collection behavior	0.8034	0.3921	0.8486	0.3184	0.8591	0.2614

Based on the results shown in Table 4, we see that DBMGN using pre-collection behavior alone could achieve relative AUC improvement of 3.5%, 1.8%, 2.1% across three datasets compared to DIN, which uses *voucher_id* but without related user behavior. Moreover, adding the post-collection behavior could further improve the overall performance by 0.7%, 0.2%, 0.9% compared to using the pre-collection only. Those pre-collection user behavior usually reflects a user's specific interest during a voucher campaign, and the post-collection could further reflect the interest stimulated by the current collected voucher. The results indicate that DBMGN is able to capture user's interest in both before and after voucher collection for better VRR prediction.

5.4 Analysis of User Activeness (Q3)

Experiments are performed to see how DMBGN is affected by users' activeness in terms of the historical UVG sequence length. Region-A which is the most active region is taken for analysis. Fig. 4 visualizes the test AUC with respect to various historical UVG sequence lengths from 0 to 6 (the max historical UVG sequence length in our system setting). Note that the test data size under each sequence length group is more than 10k, which is statistically adequate for this analysis.
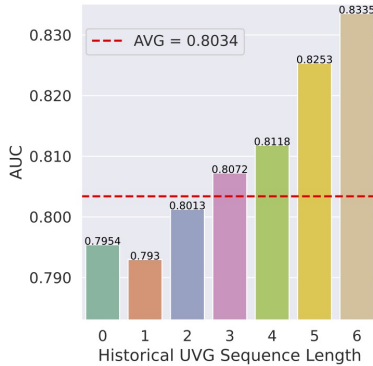

Figure 4: AUC performance on different historical UVG sequence length.

In Fig. 4, as the historical UVG sequence length increases, the corresponding AUC increases from 0.7930 to 0.8335. This observation indicates that the attention structure in DMBGN can effectively capture users' long-term interest in voucher usage. For the groups with historical sequence length larger than 1, where the attention mechanism actually matters, the performance is better than those with smaller historical length. It is interesting to observe that the AUC of the group with length 1 is slightly lower than that of the group with sequence 0. A possible explanation is that the users from the former group are also inactive in voucher usage, and only one single redeemed session might not be sufficient to reflect the user's voucher preference, but nevertheless introduces noise.

5.5 User-behavior Embedding Visualization (Q4)

To show the effectiveness of knowledge distilling by the Higher-order Graph Neural Networks, we randomly sample the sessions of five *voucher_ids* from dataset Region-B, and visualize the corresponding learned user-behavior embedding vectors $e_{UVG,b}$ from UVG in 2 dimension using t-SNE [29] algorithm in Fig. 5. All non-redeemed sessions from those *voucher_ids* are marked in red color, while the redeemed sessions with the same *voucher_id* are marked with an individual color. It shows that the learned embeddings can well discriminate redeemed and non-redeemed sessions. It also illustrates that sessions with the same color are mostly located in the same cluster. One exception is Voucher 4 cluster, which is divided into two sub-clusters. A further analysis shows that the split clusters correspond to users of different activeness levels. Specifically, in the purple sub-cluster, 93.3% users have greater than or equal to 19 total items related to user behavior while in the green sub-cluster, 83.1% users have less than 19 items. Overall, Fig. 5 indicates that the learned embeddings from DMBGN can well represent the users' voucher-related behavior.
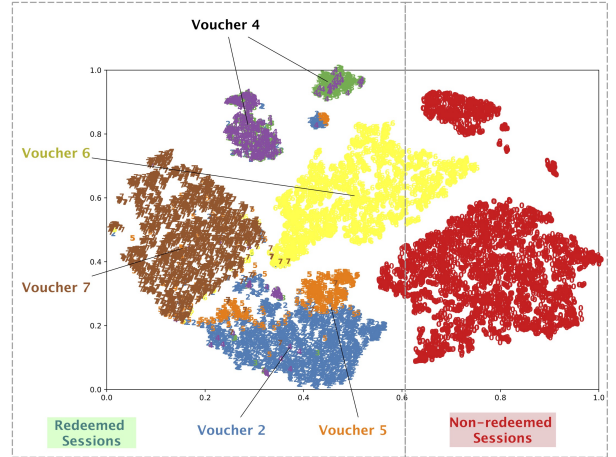

Figure 5: Visualization of learned user-behavior embedding vectors $e_{UVG,b}$ from UVG. Red color represents non-redeemed sessions, and the rest colors represent redeemed sessions with respect to different voucher ids.

6 CONCLUSION AND FUTURE WORK

In this paper, we propose a voucher redemption prediction (VRR) model: Deep Multi-Behavior Graph Networks (DMBGN). It considers the historical user behavior happening both before and after

voucher collection to make better inference. To capture the user-voucher-item structural relationships, a User-Behavior Voucher Graph (UVG) is constructed, and Higher-order Graph Neural Networks are applied to extract the structural features reflecting interactions between voucher and user behavior. An attention network is built on top of the UVG sequences to mine user's long-term voucher usage preference. Extensive experiments in three large-scale datasets are conducted. The results show that DMBGN achieves 10% to 16% relative uplift in AUC compared to DNN, as well as 2% to 4% relative uplift in AUC compared to DIN. This demonstrates that DMBGN is effective for the VRR prediction task.

There are several potential future works for exploration. In a voucher campaign, there may exist multiple vouchers to be distributed simultaneously or within a short time interval. The influence of different vouchers on user behavior are intertwined. More sophisticated graph structures are needed to capture voucher-voucher cooperative or conflicting relationships. Besides, the applicable condition of a voucher may have restrictions over specific shops, brands or categories. The influence of the voucher usage scopes on the user behavior is also worthwhile to be explored.

REFERENCES

[1] Shipra Agrawal and Nikhil R Devanur. 2015. Linear contextual bandits with knapsacks. *arXiv preprint arXiv:1507.06738* (2015).
[2] Wentian Bao, Hong Wen, Sha Li, Xiao-Yang Liu, Quan Lin, and Keping Yang. 2020. Gmcm: Graph-based micro-behavior conversion model for post-click conversion rate estimation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2201–2210.
[3] Deeparnab Chakrabarty, Yunhong Zhou, and Rajan Lukose. 2008. Online knapsack problems. In *Workshop on internet and network economics (WINE)*.
[4] C. Chen, Min Zhang, Yongfeng Zhang, W. Ma, Yiqun Liu, and Shaoping Ma. 2020. Efficient Heterogeneous Collaborative Filtering without Negative Sampling for Recommendation. In *AAAI*.
[5] Qiwei Chen, Huan Zhao, Wei Li, et al. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. 1–4.
[6] 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.
[7] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 135–144.
[8] Yufei Feng, Binbin Hu, Fuyu Lv, Qingwen Liu, Zhiqiang Zhang, and Wenwu Ou. 2020. ATBRG: Adaptive Target-Behavior Relational Graph Network for Effective Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2231–2240.
[9] Keke Gai and Meikang Qiu. 2018. Optimal resource allocation using reinforcement learning for IoT content-centric services. *Applied Soft Computing* 70 (2018), 12–21.
[10] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. 2019. Neural Multi-task Recommendation from Multi-behavior Data. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 1554–1557. https://doi.org/10.1109/ICDE.2019.00140
[11] Hongyang Gao and Shuiwang Ji. 2019. Graph u-nets. In *international conference on machine learning*. PMLR, 2083–2092.
[12] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.
[13] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (Melbourne, Australia) *(IJCAI'17)*. AAAI Press, 1725–1731.
[14] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216* (2017).
[15] Wan-Hsun Hu, Shih-Hsien Tang, Yin-Che Chen, et al. 2018. Promotion recommendation method and system based on random forest. In *Proceedings of the 5th Multidisciplinary International Social Networks Conference*. 1–5.
[16] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 659–668.
[17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, et al. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017), 3146–3154.
[18] Hans Kellerer, Ulrich Pferschy, and David Pisinger. 2004. The multiple-choice knapsack problem. In *Knapsack Problems*. Springer, 317–347.
[19] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[20] Liangwei Li, Liucheng Sun, Chenwei Weng, Chengfu Huo, and Weijun Ren. 2020. Spending Money Wisely: Online Electronic Coupon Allocation based on Real-Time User Intent Detection. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2597–2604.
[21] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* (2015).
[22] 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.
[23] H. Brendan McMahan, Gary Holt, D. Sculley, Michael Young, Dietmar Ebner, et al. 2013. Ad Click Prediction: A View from the Trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Chicago, Illinois, USA) *(KDD '13)*. Association for Computing Machinery, New York, NY, USA, 1222–1230. https://doi.org/10.1145/2487575.2488200
[24] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4602–4609.
[25] Xichuan Niu, Bofang Li, Chenliang Li, Rong Xiao, Haochuan Sun, Hongbo Deng, and Zhenzhong Chen. 2020. A Dual Heterogeneous Graph Attention Network to Improve Long-Tail Performance for Shop Search in E-Commerce. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3405–3415.
[26] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.
[27] Andreas Pfadler, Huan Zhao, Jizhe Wang, Lifeng Wang, Pipei Huang, and Dik Lun Lee. 2020. Billion-scale Recommendation with Heterogeneous Side Information at Taobao. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1667–1676.
[28] Shi Qiongyu. 2020. Prediction of O2O Coupon Usage Based on XGBoost Model. In *2020 The 11th International Conference on E-business, Management and Economics*. 33–36.
[29] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html
[30] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 225–232.
[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, et al. 2018. Graph Attention Networks. arXiv:1710.10903 [stat.ML]
[32] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.
[33] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 2020. Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2377–2386.
[34] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Bo Zhang, et al. 2020. Multi-plex Behavioral Relation Learning for Recommendation via Memory Augmented Transformer Network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2397–2406.
[35] 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 the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1905–1908.
[36] Ling Yan, Wu-jun Li, Gui-Rong Xue, and Dingyi Han. 2014. Coupled group lasso for web-scale ctr prediction in display advertising. In *International Conference on Machine Learning*. PMLR, 802–810.
[37] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, et al. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.
[38] 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.