

ATBRG: Adaptive Target-Behavior Relational Graph Network for Effective Recommendation

Yufei Feng^{1*}, Binbin Hu^{2*}, Fuyu Lv¹, Qingwen Liu¹, Zhiqiang Zhang², Wenwu Ou¹

¹Alibaba Group, Hangzhou, China

²Ant Financial Services Group, Hangzhou, China

fjf649435349@gmail.com, bin.hbb@antfin.com, {fuyu.lfy, xiangsheng.lqw}@alibaba-inc.com, lingyao.zzq@antfin.com, santong.oww@taobao.com

ABSTRACT

Recommender system (RS) devotes to predicting user preference to a given item and has been widely deployed in most web-scale applications. Recently, knowledge graph (KG) attracts much attention in RS due to its abundant connective information. Existing methods either explore independent meta-paths for user-item pairs over KG, or employ graph neural network (GNN) on whole KG to produce representations for users and items separately. Despite effectiveness, the former type of methods fails to fully capture structural information implied in KG, while the latter ignores the mutual effect between target user and item during the embedding propagation. In this work, we propose a new framework named Adaptive Target-Behavior Relational Graph network (ATBRG for short) to effectively capture structural relations of target user-item pairs over KG. Specifically, to associate the given target item with user behaviors over KG, we propose the graph connect and graph prune techniques to construct adaptive target-behavior relational graph. To fully distill structural information from the sub-graph connected by rich relations in an end-to-end fashion, we elaborate on the model design of ATBRG, equipped with relation-aware extractor layer and representation activation layer. We perform extensive experiments on both industrial and benchmark datasets. Empirical results show that ATBRG consistently and significantly outperforms state-of-the-art methods. Moreover, ATBRG has also achieved a performance improvement of 5.1% on CTR metric after successful deployment in one popular recommendation scenario of Taobao APP.

CCS CONCEPTS

• Information systems → Recommender systems; • Computing methodologies → Neural networks.

KEYWORDS

Recommender System; Knowledge Graph; Graph Neural Network

* Yufei Feng and Binbin Hu contribute equally.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '20, July 25–30, 2020, Virtual Event, China

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-8016-4/20/07...\$15.00

<https://doi.org/10.1145/3397271.3401428>

ACM Reference Format:

Yufei Feng, Binbin Hu, Fuyu Lv, Qingwen Liu, Zhiqiang Zhang, 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 (SIGIR'20), July 25–30, 2020, Virtual Event, China*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3397271.3401428>

1 INTRODUCTION

In the era of information overload, recommender system (RS), which aims to match diverse user interests with tremendous resource items, are widely deployed in various online services, including e-commerce [17, 27, 32], social media [3, 33] and news [4, 21]. Traditional recommendation methods, e.g., matrix factorization [11], mainly learn an effective preference prediction function using historical user-item interaction records. Despite effectiveness, these methods suffer from cold-start problem due to data sparsity. With the rapid development of web services, some approaches [8, 9] are proposed to incorporate various auxiliary data for improving recommendation performance.

Recently, knowledge graph (KG), which is flexible to model comprehensive auxiliary data, has attracted increasing attention in RS [2, 20, 22–25]. Generally, KG stores external heterogeneous knowledge in the ternary form $\langle \text{head entity}, \text{relation}, \text{tail entity} \rangle$, corresponding to attribute (e.g., $\langle \text{Blouse}, \text{Category}, \text{Shirt} \rangle$) or relationship (e.g., $\langle \text{Shirt}, \text{Audience}, \text{Girl} \rangle$) of entities. Due to its abundant information, current recommender systems mainly aim to incorporate KG to enrich representations of users and items and promote the interpretability of recommendations.

Though with great improvements, it remains challenging to effectively integrate such heterogeneous information for recommendation. Roughly speaking, state-of-the-art KG based recommendation methods mainly fall into two groups, path based and graph neural network (GNN) based methods. Path based methods [24] infer user preference by exploring multiple meta-paths for target user-item pairs over KG, which typically requires domain knowledge. More importantly, this type of methods ignores rich structural information implied in KG, and thus cannot sufficiently characterize the underlying relationships between given target user and item. As illustrated in Fig. 1, these methods essentially overlook the strong relationships between *Blouse* and *Dress*, since each extracted path is modeled independently.

Inspired by the recently emerging graph neural networks, several GNN based methods [22, 23] have been proposed and provide strong performance by explicitly modeling high-order connectivities in KG. Nevertheless, these methods still suffer from three

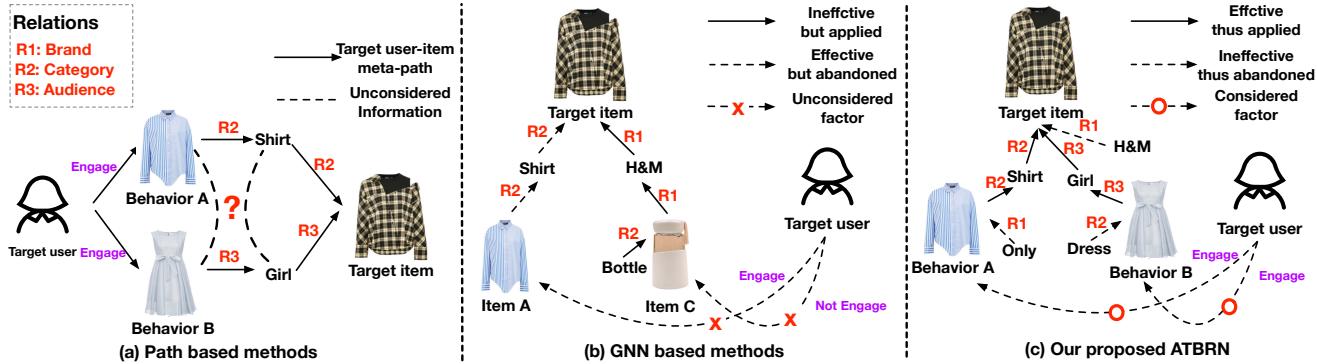


Figure 1: The comparison of our proposed framework ATBRG with previous models. (a) & (b) indicate the limitations of path based and GNN based methods, respectively, while (c) shows the superiority of ATBRG .

limitations: **(L1)** These methods mainly apply GNN to enrich the representation of target user and item separately by aggregating their own original neighbors in the KG, and thus fail to capture their mutual influence during the procedure of information aggregation. As shown in Fig. 1 (b), current GNN based methods tend to produce representation for target item through aggregating its neighbors without considering target user’s interests (history behaviors). Subsequently, some unnecessary information (*i.e.*, *Cup*) is involved in target item’s refined embedding, which may harm recommendation performance; **(L2)** The KG in the real-world industrial scenario is extremely large-scale, where one entity can be linked with up to millions of items. Existing works mainly employ the random sampling on the neighbors beforehand, which may lose latent critical information for the specific target user and item. As shown in Fig. 1 (b), some neighbors (*i.e.*, *Shirt*) are abandoned by the random sampling strategy, while they are usually informative during aggregation since the target user has engaged with them; **(L3)** Most of these methods neglect rich relations among user behaviors over KG, while some works [5, 31] have demonstrated that capturing the relations among user behaviors is also beneficial for expressing user preference.

To address above limitations, we aim to distill the original over-informative KG into recommendation in a more effective way, which is expected to satisfy the following key properties: **(1) Target-behavior:** we hang on the novel insight that an effective KG base recommendation should produce semantic sub-graph to adapt for each target user-item pair, with the aim of capturing the underlying mutual effect characterized by KG **(L1)**; **(2) Adaptive:** distinct from random sampling on the whole KG, our idea is to follow the adaptive principle for the sub-graph construction, which adaptively preserves useful information connecting user behaviors and target item over the KG, driving our model to provide more effective recommendation **(L2)**; **(3) Relational:** the model architecture should be designed to relation-aware in order to consider the rich relations among user behaviors and target item over KG **(L3)**. For convenience, given a target user-item pair, we call the relational graphical structure bridging user behaviors (*i.e.*, historical click records) with the target item as *adaptive target-behavior relational graph* (shown in Fig. 1(c)). Propagating user preference on such a

relational structure potentially takes full advantage of the mutual effect for target user-item pair, as well as comprehensively captures the structural relations derived from KG.

In this paper, by integrating above main ideas together, we propose a new framework named **Adaptive Target-Behavior Relational Graph Network (ATBRG)**, which is comprised of two main parts: (1) Graph construction part. To extract the effective relational sub-graph of for target the user-item pair over the KG adaptively, we propose the *graph connect* and *graph prune* techniques. Firstly, we explore multiple layer neighbors over KG for target item and each item in user behavior, respectively. Among these entity sets, we *connect* entities which appear in multiple entity sets and *prune* entities belonging to only one entity set. Subsequently, we construct the adaptive target-behavior relational graph, which characterizes the structural relations among user behaviors and target item over the KG. (2) Model part. Considering structural relations derived from KG, we technically design the relation-aware extractor layer, which employs relation-aware attention mechanism to aggregate structural knowledge over the relational graph for each user behavior and target item. Afterwards, we introduce the representation activation layer to activate the relative relational representations of user behavior *w.r.t.* that of target item.

The main contributions of our work are summarized as follows:

- To effectively characterize structural relations between the given target user and item, we propose to extract an adaptive target-behavior relational graph, where the graph connect and graph prune strategies are developed to adaptively build relations between user behaviors and target item over KG.
- We propose a novel framework ATBRG, a well-designed graph neural network based architecture to learn relational representations of user behaviors and target item over the extracted sub-graph. Moreover, we equip it with relation-aware extractor layer and representation activation layer for emphasizing rich relations for interaction in KG.
- We perform a series of experiments on a benchmark dataset from Yelp and an industrial dataset from Taobao App. Experimental results demonstrate that ATBRG consistently and significantly outperforms various state-of-the-art methods. Moreover, ATBRG has been successfully deployed in

one popular recommendation scenario of Taobao APP and gained a performance improvement of 5.1% on CTR metric.

2 RELATED WORK

In this section, we review the most related studies in behavior based and knowledge aware recommendation.

2.1 Behavior based recommendation

In the early stage of recommendation, researchers focus on recommending a suitable list of items based on historical user-item interaction records. In particular, a series of matrix factorization based methods [11] have been proposed to infer user preference towards items through learning latent representations of users and items. Due to the ability of modeling complex interaction between users and items, deep neural network based methods (e.g., Youtubenet [3], DeepFM [7]) are widely adopted in industrial recommender systems, and reveal the remarkable strength of incorporating various context information (e.g., user profile and item attributes).

In the online e-commerce systems, we are particularly interested in user's historical behaviors, which implies rich information for inferring user preference. Hence, how to effectively characterize the relationships between user behaviors and target item remains a continuous research topic. DIN [32] adaptively learns the representation of user interests from historical behaviors w.r.t. the target item by the attention mechanism. Inspired by DIN, the majority of following up works inherit this kind of paradigm. GIN [12] mines user intention based on co-occurrence commodity graph in the end-to-end fashion. ATRANK [30] proposes an attention-based behavior modeling framework to model users' heterogeneous behaviors. DIEN [31] and SDM [14] devote to capturing users' temporal interests and modeling their sequential relations. DSIN [5] focuses on capturing the relationships of users' inter-session and intra-session behaviors. MIMN [15] and HPMN [16] apply the neural turing machine to model users' lifelong sequential behaviors. Besides these improvements, knowledge graph, consisting of various semantics and relations, emerges as an assistant to describe relationships between user behaviors and target item.

2.2 Knowledge Aware Recommendation

As a newly emerging direction, knowledge graph is widely integrated into recommender systems for enriching relationships among user behaviors and items. A research line utilizes KG aware embeddings (e.g., structural embeddings [28] and semantics embeddings [21]) to enhance the quality of item representations. These methods conduct multi-task learning within two tasks of recommendation and KG completion and share the embeddings, and thus can hardly take full advantage of high-order information over KG. On the contrary, several efforts [8, 24] have been made to explore different semantic path (meta-path) connecting target users and items over KG, and then learn prediction function through multiple path modeling. More recently, some works [26] propose to exploit reinforcement learning to explore useful path for recommendation. Despite effectiveness, the path based method ignores rich structural information implied in KG since each extracted path is modeled independently.

Recently, graph neural network has shown its potential in learning accurate node embeddings with the high-order graph topology. Taking advantages of information propagation, RippleNet [20] propagates users' potential preferences and explores their hierarchical interests over KG, while KGNC-LS [22] and KGAT [23] perform embedding propagation by stacking multiple KG aware GNN layers. Although GNN based methods have achieved performance improvement to some extent, they do not take mutual influence between target user behaviors and item into consideration in the procedure of information aggregation. Moreover, the exponential neighborhood expansion over graph extremely increases the complexity of the system.

3 PRELIMINARY

In a recommendation scenario (e.g., e-commerce and news), we typically have a series of historical interaction records (e.g., purchases and clicks) between users and items. Let \mathcal{U} denote a set of users and \mathcal{I} denote a set of items, we denote interaction records as $\mathcal{H} = \{(u, i, \mathcal{B}_{ui}, y_{ui}) | u \in \mathcal{U}, i \in \mathcal{I}\}$. Here, $\mathcal{B}_{ui} \subset \mathcal{I}$ represents historical behaviors (i.e., item list) for user u when recommending item i and $y_{ui} \in \{0, 1\}$ is the implicit feedback of user u w.r.t. item i , where $y_{ui} = 1$ when $\langle u, i \rangle$ interaction is observed, and $y_{ui} = 0$ otherwise. In the real-world industrial recommender systems, each user u is associated with a user profile x_u consisting of sparse features (e.g., user id and gender) and numerical features (e.g., user age), while each item i is also associated with a item profile x_i consisting of sparse features (e.g., item id and brand) and numerical features (e.g., price).

In order to effectively incorporate auxiliary information of items (i.e., item attributes and external knowledge) into recommendation, we frame our recommendation task over knowledge graph, which can be defined as follows:

DEFINITION 1. *Knowledge Graph.* A KG is defined as a directed graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$ with an entity set \mathcal{E} and a relation set \mathcal{R} . Each triplet $(h, r, t) \in \mathcal{T}$ denotes a fact that there is a relationship r from head entity h to tail entity t , where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$.

For example, $\langle \text{Blouse}, \text{Category}, \text{Shirt} \rangle$ states the fact that *Blouse* belongs to the *Shirt Category*. To bridge knowledge graph with recommender system, we adopt a item-entity alignments function $\phi : \mathcal{I} \rightarrow \mathcal{E}$ to align items with entities in KG.

Many efforts, especially GNN based methods, have been made to leveraging KG for better recommendation. While, most of these works overlook the mutual effect between target user and item when exploiting structural information derived from KG. To effectively distill structural knowledge through KG based GNN for item recommendation, we particularly investigate into the external knowledge connecting user behaviors and target item in KG, which can reveal semantic context for user-item interactions. Formally, we define such context information as follows:

DEFINITION 2. *Adaptive Target-Behavior Relational Graph.* Given a target user-item pair $\langle u, i \rangle$ and corresponding user behaviors \mathcal{B}_{ui} , an adaptive target-behavior relational graph w.r.t. $\langle u, i \rangle$ is defined as a sub-graph extracted from the original KG, connecting user behavior \mathcal{B}_{ui} and target item i .

Table 1: Notations.

Notations	Description
\mathcal{U}, \mathcal{I}	the set of users and items, respectively
\mathcal{H}	the set of historical interaction records
y_{ui}, \hat{y}_{ui}	the label and the predicted probability
$x_u, x_i, \mathcal{B}_{ui}$	user profile, item profile and user behaviors
i_b	the specific item in user behavior
$\mathcal{G}, \mathcal{G}_{ui}$	the knowledge graph and adaptive target-behavior relational graph, respectively
$\mathcal{E}, \mathcal{R}, \mathcal{T}$	the set of entity, relation and triples in knowledge graph, respectively
$\langle h, r, t \rangle$	the specific triple in knowledge graph
$\mathbf{x}_u, \mathbf{x}_i, \mathbf{x}_e, \mathbf{x}_r$	the embedding of user u , item i , entity e and relation r , respectively
$\mathcal{N}_h^{(l)}$	the neighbors set in l -th relation-aware extractor layer for entity h
$\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_{i_b}$	the relational representation for target item i and each item $i_b \in \mathcal{B}_{ui}$, respectively
$\tilde{\mathbf{x}}_u$	the final representation of user u

Given the above preliminaries, we now formulate the recommendation task to be addressed in this paper:

DEFINITION 3. Task Description. Given a knowledge graph \mathcal{G} with historical interaction records \mathcal{H} , for each user-item pair $\langle u, i \rangle$, we aim to predict probability \hat{y}_{ui} that user u would click item i .

4 THE PROPOSED FRAMEWORK

In this section, we introduce our proposed framework ATBRG, which aims to take full advantage of knowledge graph for recommendation. The framework is shown in Fig. 2, which is composed of two modules: (1) To effectively extract structural relational knowledge for recommendation, we propose to construct the adaptive target-behavior relational graph for the given target user-item pair over knowledge graph, where the *graph connect* and *graph prune* techniques help mine high-order connective structure in an automatic manner; (2) To jointly distill such a relational graph and rich relations among user behaviors in an end-to-end framework, we elaborate on the model design of ATBRG, which propagates user preference on the sub-graph with relation-aware extractor layer and representation activation layer. The key notations we will use throughout the article are summarized in Table 1.

4.1 Graph Construction

A major novelty of our work is to effectively explore adaptive target-behavior relational graph for improving the modeling of the interaction. In this part, we introduce the strategy to construct the adaptive target-behavior relational graph with the proposed *graph connect* and *graph prune* techniques. To model the relationship between the given target user-item over KG, previous works either extract different paths through random walk [24], or directly leverage the neighbors of target item over the original KG [22, 23]. Unfortunately, the first strategy neglects the structural relational

information of the KG, while the second ignores the mutual effect between user behaviors and target item. Hence, we argue that above two strategies only achieve the suboptimal performance for recommendation.

Intuitively, the reasons driving a user to click a target item maybe implied by his/her historical behaviors, which is expected to guide our model to adequately aggregate useful information over external KG in an automatic manner. To distill the structural relational information over the KG in a more effective way, we propose to construct the adaptive relational graph w.r.t. user behaviors and target item. The procedure of the graph construction is clearly presented in the Algorithm 1 and left part of Fig. 2. Specifically, given a target user-item pair $\langle u, i \rangle$, we firstly exhaustively search the multi-layer entity neighbors for user behaviors \mathcal{B}_{ui} and target item i over the KG, and restore the paths connecting the entity and item into \mathcal{G}_{ui} (lines 1-6). Through this, we connect the user behaviors and target item by multiple overlapped entities. Afterwards, for the entities in \mathcal{G}_{ui} , we prune the entities which do not connect different items. (lines 7-16). Finally, we get the relational graph \mathcal{G}_{ui} for user u and target item i , which describes the structural relations for $\langle u, i \rangle$ over the KG.

Algorithm 1 Graph construction

Input: Target item i ; User behavior \mathcal{B}_{ui} ; Knowledge graph \mathcal{G} ;
Output: \mathcal{G}_{ui} : Adaptive target-behavior relational graph for $\langle u, i \rangle$;

```

1: for item  $v \in [i, \mathcal{B}_{ui}]$  do:
2:   for entity  $e \in \phi(v)$  do:
3:     Construct path  $p = (e, r_k, e_k, \dots, e_1, r_1, v)$ ;
4:      $\mathcal{G}_{ui}[\text{entity}] \leftarrow \mathcal{G}_{ui}[\text{entity}] \cup p$ ;      ▷ Graph connect.
5:   end for
6: end for
7: for entity  $e \in \mathcal{G}_{ui}$  do:
8:   New item hash set  $s$ ;
9:   for path  $p \in \mathcal{G}_{ui}[e]$  do:
10:    Collect item  $v$  on the path;
11:     $s \leftarrow s \cup v$ ;
12:   end for
13:   if  $s.\text{size} = 1$  then:                                ▷ Graph prune.
14:     Prune  $e$  in  $\mathcal{G}_{ui}$ ;
15:   end if
16: end for

```

4.2 Model Architecture

After obtaining the adaptive target-behavior relational graph derived from the KG, we continue to study how to produce predictive embeddings for target user-item pairs through propagating user preference over such a sub-graph. As shown in the right part of Fig. 2, the model architecture of our proposed ATBRG is composed of four layers: 1) Embedding layer, which transforms high-dimensional sparse features into low-dimensional dense representations; 2) Relation-aware extractor layer, which produces knowledge aware embeddings for user behaviors and target item by aggregating structural relational information over adaptive target-behavior relational graph; 3) Representation activation layer, which activates the relative relational representations of user behaviors w.r.t. that

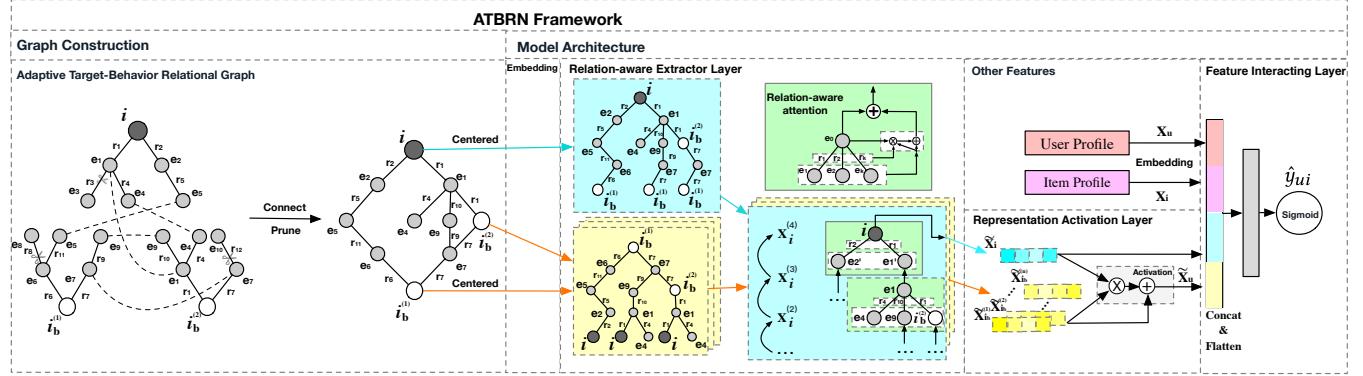


Figure 2: Overview of the proposed ATBRG framework. Overall, ATBRG consists of two parts, graph construction and model architecture.

of target item. 4) Feature interaction layer, which combines the user and item profile with the activated relational representation of user behaviors and target item for interaction.

4.2.1 Embedding Layer. As mentioned above, users and items in real-world recommendation scenario are both associated with abundant profile information in the form of sparse and dense features. Hence, we set up a embedding layer to parameterize users and items as vector representations, while preserving the above profile information. Formally, giving a user u , we have corresponding raw feature space x_u , comprised of sparse feature space x_u^s and dense feature space x_u^d . For sparse features, following [3, 5, 31, 32], we embed each feature value into d dimensional dense vector, while dense feature can be standardized or batch normalization to ensure normal distribution. Subsequently, each user u can be represented as $x_u \in \mathbb{R}^{|x_u^s| \times d + |x_u^d|}$, where $|x_u^s|$ and $|x_u^d|$ denotes the size of sparse and dense feature space of user u , respectively. Similarly, we represent each item i as $x_i \in \mathbb{R}^{|x_i^s| \times d + |x_i^d|}$. Moreover, each entity e and relation r in the adaptive target-behavior relational graph can also be embedded as $x_e \in \mathbb{R}^d$ and $x_r \in \mathbb{R}^d$ ¹.

4.2.2 Relation-aware Extractor Layer. This layer is designed to effectively and comprehensively distill the structural relational information from the extracted sub-graph. Previous works [10, 19] neglect relational edges during aggregation, which play essential roles in real-world settings. In our scenario, a user u may *click* or *buy* the same item i , while the relations *click* and *buy* obviously indicate the different preference of user u towards item i . Therefore, we elaborately build the relation-aware extractor layer to adequately exploit rich structural relational information in KG in the consideration of various relation between entities.

Based on the above discussions and inspired by the study [6], we stack the relation-aware extractor layer by layer in order to recursively propagates the embeddings from an entity's neighbors to refine the entity's embedding in KG. Specifically, for each item (*i.e.*, item $i_b \in \mathcal{B}_{ui}$ or target item i), we will regard it as the center node and aggregate information over the extracted sub-graph

\mathcal{G}_{ui} through relation-aware aggregation. Given an entity ² h in extracted relational sub-graph \mathcal{G}_{ui} , let $\mathcal{N}_h^{(l)} = \{(r, t) | (h, r, t) \in \mathcal{G}_{ui}\}$ to denote the neighbors set in l -th layer and $x_h^{(l)}$ to denote the representation of entity h in the l -th layer. We implement the l -th relation-aware aggregation layer as follows,

$$\alpha^{(l)}(h, r, t) = \frac{\exp(\mathbf{x}_r \mathbf{W}_\alpha f(x_h^{(l)} \oplus x_t^{(l)}))}{\sum_{(r', t') \in \mathcal{N}_h^{(l)}} \exp(\mathbf{x}_{r'} \mathbf{W}_\alpha f(x_h^{(l)} \oplus x_{t'}^{(l)}))}, \quad (1)$$

$$\mathbf{x}_h^{(l+1)} = \mathbf{x}_h^{(l)} \oplus \sum_{(r, t) \in \mathcal{N}_h^{(l)}} \alpha^{(l)}(h, r, t) \mathbf{x}_t^{(l)}.$$

Here, $f(\mathbf{x})$ and \mathbf{W}_α denote the single layer perceptron and attentive matrix in l -th layer, respectively. And \oplus denotes the concatenation operation. Relation-aware extractor layer is stacked layer by layer to propagate user preference over KG. Subsequently, each entity e in sub-graph \mathcal{G}_{ui} can be denoted as $x_e^{(L)}$ after L relation-aware extractor layer.

Given a target user-item pair $\langle u, i \rangle$ and the corresponding adaptive target-behavior relational graph \mathcal{G}_{ui} , the knowledge aware representation of target item i can be denoted as $\tilde{\mathbf{x}}_i = \mathbf{x}_i^{(L)}$, where $\phi(i) = e$. Similarly, we also obtain relational representation set $\{\tilde{\mathbf{x}}_{i_b} = \mathbf{x}_{i_b}^{(L)} | \phi(i_b) = e\}_{i_b \in \mathcal{B}_{ui}}$ for user behaviors.

4.2.3 Representation Activation Layer. Intuitively, user behaviors contribute differently to the final prediction. For example, the behavior *shirt A* is more informative than *shoe B* when the target item is *shirt C*. For this purpose, we set a representation activation layer to place different importance on relational representation of user behaviors $\{\tilde{\mathbf{x}}_{i_b}\}_{i_b \in \mathcal{B}_{ui}}$. Specifically, we apply the vanilla attention mechanism [1] to activate representations of user behaviors that are more related to target item, calculated as follows,

$$\beta(u, i, i_b) = \frac{\exp(\tilde{\mathbf{x}}_{i_b} \mathbf{W}_\beta \tilde{\mathbf{x}}_i))}{\sum_{i'_b \in \mathcal{B}_{ui}} \exp(\tilde{\mathbf{x}}_{i'_b} \mathbf{W}_\beta \tilde{\mathbf{x}}_i)}, \quad (2)$$

$$\tilde{\mathbf{x}}_u = \sum_{i'_b \in \mathcal{B}_{ui}} \beta(u, i, i_b) \tilde{\mathbf{x}}_{i'_b},$$

¹Entities and relations in KG only have the sparse id features

²For convenience, we omit the subscript ui in this part.

Table 2: Statistics of datasets

	Description	Taobao	Yelp
User-Item Interaction	#Users	2.2×10^8	4.5×10^4
	#Items	1.1×10^8	4.5×10^4
	#Interactions	7.2×10^9	1.0×10^6
Knowledge Graph	#Entities	1.4×10^7	8.3×10^4
	#Relations	34	35
	#Triplets	3.8×10^{10}	1.6×10^6
	#Max neighbor depth	3	1

where \mathbf{W}_β is the attentive matrix in representation activation layer.

4.2.4 Feature Interaction Layer. Until now, given a target user-item pair $\langle u, i \rangle$, we have the profile embeddings for user u and item i , and knowledge aware embedding from adaptive target-behavior relational graph for user behaviors and target item. We combine the four embedding vectors into a unified representation and employ Multiple Layer Perceptron (MLP) for better feature interaction [5, 15, 30–32].

$$\hat{y}_{ui} = \sigma(f(f(f(\mathbf{x}_u \oplus \mathbf{x}_i \oplus \tilde{\mathbf{x}}_u \oplus \tilde{\mathbf{x}}_i))), \quad (3)$$

where $\sigma(\cdot)$ is the logistic function and \hat{y}_{ui} represents the prediction probability of the user u to click on the target item i .

4.2.5 Loss Function. We reduce the task to a binary classification problem and use binary cross-entropy loss function defined as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{(u,i) \in \mathbb{D}} (y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui})) \quad (4)$$

where \mathbb{D} is the training dataset and $y_{ui} \in \{0, 1\}$ represents whether the user u clicked on the target item i .

5 EXPERIMENTS

In this section, we perform a series of experiments on two real-world datasets, with the aims of answering the following research questions:

- **RQ1:** How does our proposed model ATBRG perform compared with state-of-the-art methods on the recommendation task?
- **RQ2:** How do different experimental settings (i.e., depth of graph, aggregator selection, etc.) influence the performance of ATBRG?
- **RQ3:** How does ATBRG provide effective recommendation intuitively?

5.1 Experimental Setup

5.1.1 Datasets. We conduct extensive experiments on two real-world datasets: industrial dataset from Taobao and benchmark dataset from Yelp.

- **Taobao**³ dataset consists of click logs from 2019/08/22 to 2019/08/29, where the first one week's samples are used for training and samples of the last day are for testing. Moreover, Taobao dataset also contains user profile (e.g., id and age),

³www.taobao.com.

item profile (e.g., id and category) and up to 10 real-time user behaviors⁴.

- **Yelp**⁵ dataset records interactions between users and local business and contains user profile (e.g., id, review count and fans), item profile (e.g., id, city and stars) and up to 10 real-time user behaviors. For each observed interaction, we randomly sample 5 items that the target user did not engage with before as negative instances. For each user, we hold the latest 30 instances as the test set and utilizes the remaining data for training.

Besides user behaviors, following [13, 18, 29], we construct item knowledge for Taobao (e.g., category, parent and style). Also, for Yelp dataset, KG is organized as the local business information (e.g., location and category). The detailed descriptions of the two datasets are shown in Table 2. Note that the volume of Taobao dataset is much larger than yelp, which brings more challenges.

5.1.2 Baselines. We compare our ATBRG with three kinds of representative methods: feature based methods (*i.e.*, YoutubeNet and DeepFM) mainly utilizing raw features derived from user and item profile, behavior based methods (*i.e.*, DIN, DIEN and DSIN) capturing user's historical behaviors and knowledge graph (KG) based methods (*i.e.*, RippleNet, KGAT and KPRN) benefiting from knowledge graphs in recommendation. The comparison methods are given below in detail:

- **YoutubeNet** [3] is a standard user behavior based method in the industrial recommender system.
- **DeepFM** [7] combines factorization machine and deep neural network for recommendation.
- **DIN** [32] locates related user behaviors *w.r.t.* target item by using attention mechanism.
- **DIEN** [31] models users temporary interests and the interest evolving process via GRU with attention update gate.
- **DSIN** [5] models user's session interests and the evolving process with self-attention mechanism and Bi-LSTM.
- **RippleNet** [20] propagates user's potential preferences over the set of knowledge entities.
- **KPRN** [24] is a typical path based recommendation method, which extracts qualified path to between a user with an item.
- **KGAT** [23] is a state-of-the-art KG-based recommendation methods, which employs GNN on KG to generate representations of users and items, respectively.

5.1.3 Evaluation Metrics. We adopt area under ROC curve (**AUC**) to evaluate the performance of all methods. Larger AUC indicates better performance. Besides, we also present the relative improvement (**RI**) *w.r.t.* AUC of our model achieves over the compared models, which can be formulated as:

$$RI = \frac{|AUC(model) - AUC(base)|}{AUC(base)} * 100\%, \quad (5)$$

where $|.|$ is the absolute value, *model* refers to our proposed framework ATBRG and *base* refers to the baseline. Note that **0.001** improvement *w.r.t.* AUC is remarkable in industrial scenario (*i.e.*, Taobao dataset).

⁴Real-time user behaviors means user behaviors before this action occurs.

⁵www.yelp.com/dataset.

Table 3: Overall performance comparison w.r.t. AUC (bold best; underline: runner-up).

Model	Taobao		Yelp [†]	
	AUC	RI	AUC	RI
Youtubenet	0.6017	+2.72%	0.7109	+26.00%
DeepFM	<u>0.6037</u>	+2.38%	<u>0.7334</u>	+22.14%
DIN	0.6058	+2.03%	0.7520	+19.12%
DIEN	0.6061	+1.97%	0.7581	+18.16%
DSIN	0.6073	+1.77%	0.7774	+15.23%
RippleNet	0.5975	+3.44%	0.7324	+22.31%
KGAT	0.6062	+1.96%	0.7876	+13.73%
KPRN	<u>0.6096</u>	+1.39%	<u>0.8260</u>	+8.45%
ATBRG	0.6181*	-	0.8958*	-

[†] Note that the relative improvement on the public Yelp dataset is much higher than the industrial Taobao dataset, since the negative samples of the public Yelp dataset are generated by random sampling and thus easier to distinguish.

5.1.4 Implementation. We implement all models in Tensorflow 1.4. Moreover, for fair comparison, pre-training, batch normalization and regularization are not adopted in our experiments. For RippleNet, we set the max depth of ripple as 3. For KGAT, the max neighbour depth of target user and item is set to 4 and 3, respectively. For KPRN, the max number of extracted paths over the knowledge graph are set to 50. For ATBRG, the max neighbor depth of the item is set to 3. For all models, We employ random uniform to initialize model parameters and adopt Adagrad as optimizer using a learning rate of 0.001. Moreover, embedding size of each feature is set to 4 and the architecture of MLP is set to [512, 256, 128]. We run each model three times and reported the mean of results.

5.1.5 Significance Test. For Experimental results in Tables 4, 5, 6 and 7, we use “*” to indicate that ATBRG is significantly different from the runner-up method based on paired t-tests at the significance level of 0.01.

5.2 Performance Comparison (RQ1)

We report the AUC comparison results of ATBRG and baselines on two datasets in Table 3. The major findings from the experimental results are summarized as follows:

- Feature based methods (*i.e.*, Youtubenet and DeepFM) achieve relatively pool performance on two datasets. It indicates that handcrafted feature engineering is insufficient to capture the complex relations between users and items, further limiting performance. Moreover, DeepFM consistently outperforms Youtubenet across all cases, since it employs FM part for better feature interaction.
- Compared to feature based methods, the performance of behavior based methods (*i.e.*, DIN, DIEN and DSIN) verifies that incorporating historical behaviors is beneficial to infer user’s preference. Among them, DSIN achieves the best performance on both datasets due to integration of user’s session interests.
- Generally, KG based methods (*i.e.*, RippleNet, KGAT, KPRN) achieve better performance than behavior based methods

Table 4: Effect of the representation activation layer and relation-aware mechanism.

Model	Taobao		Yelp	
	AUC	RI	AUC	RI
ATBRG _{w/o RAM}	0.6157	+0.38%	0.8858	+1.12%
ATBRG _{w/o RAL}	0.6125	+0.91%	0.8940	+0.20%
ATBRG	0.6181*	-	0.8958*	-

Table 5: Effect of the depth of neighbor.

Model [†]	Taobao		Yelp	
	AUC	RI	AUC	RI
ATBRG _{1/0}	0.6054	+2.09%	0.7523	+19.07%
ATBRG _{3/1}	0.6143	+0.61%	0.8958*	-
ATBRG _{5/2}	0.6181*	-	-	-
ATBRG _{7/3}	0.6163	+0.29%	-	-

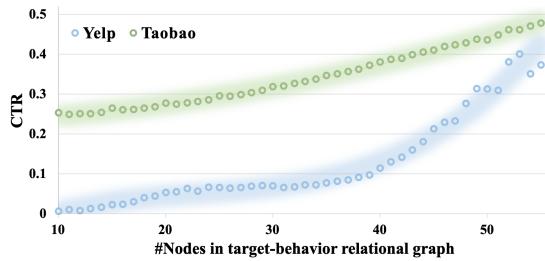
[†]ATBRG_{m/n} means ATBRG explores m layers of neighbors over the extracted sub-graph, which corresponds to n layers of neighbors in original KG.

in most cases, which indicates the effectiveness of knowledge graph for capturing underlying interaction between users and items. However, RippleNet underperforms other baselines on both datasets. One possible reason is that RippleNet ignores user’s short-term interest implied in historical behaviors. Moreover, KPRN generally achieves remarkable improvements in most cases. It makes sense since reasonable and explainable target user-item paths extracted from KG are helpful to improve recommendation performance.

- ATBRG consistently yields the best performance on both datasets. In particular, ATBRG improves over the best baseline w.r.t. AUC by 1.39%, and 8.45% on Taobao and Yelp dataset, respectively. By stacking multiple GNN layers, ATBRG is capable of exploring rich structural and relational information over KG, while KPRN only models each extracted path independently. This verifies the importance of capturing both semantics and topological structures derived from KG for recommendation. Besides, compared with KGAT, which only represents target user and item separately by aggregating their own neighbors over the original KG, ATBRG achieves better performance for the following two reasons: 1) ATBRG considers the mutual effect between the given user behaviors and target item by constructing the adaptive relational sub-graph for them. Propagating on such a sub-graph can better capture the structural relations between user behaviors and target item and further explore potential reasons driving the user to click the target item; 2) ATBRG integrates relations when aggregating the entities by the relation-aware attention mechanism, and creatively produces the relational representations over the extracted sub-graph for each user behavior and target item.

Table 6: Effect of different aggregators.

Model	Taobao		Yelp	
	AUC	RI	AUC	RI
ATBRG _{concat}	0.6131	+0.64%	0.8901	+0.65%
ATBRG _{sum}	0.6133	+0.78%	0.8906	+0.58%
ATBRG _{sa}	0.6145	+0.58%	0.8928	+0.33%
ATBRG _{nl}	0.6159	+0.35%	0.8946	+0.13%
ATBRG	0.6181*	-	0.8958*	-

**Figure 3: Impact of the number of nodes in adaptive target-behavior relational graph w.r.t. CTR.**

5.3 Study of ATBRG (RQ2)

In this section, we perform a series of experiments to better understand the traits of ATBRG, including well-designed components (*e.g.*, relation-aware mechanism and representation activation layer) and key parameter settings (*i.e.*, neighbor depth and aggregator).

5.3.1 Effect of Relation-aware Mechanism and Representation Activation Layer. ATBRG provides a principled way to characterize various relations in KG and user behaviors to enhance recommendation performance. To examine the effectiveness of relation-aware mechanism and representation activation layer, we prepare three variants of ATBRG:

- **ATBRG_{w/o RAM}**: The variant of ATBRG, which removes the relation-aware mechanism (Eq. 1).
- **ATBRG_{w/o RAL}**: The variant of ATBRG, which removes the representation activation layer (Eq. 2).

The AUC comparison results of ATBRG with its variants are shown in Table 4. We have the following two observations:

- It is clear that the performance of ATBRG degrades without the relation-aware mechanism on both datasets (*i.e.*, ATBRG > ATBRG_{w/o RAM}). It demonstrates that different relations in KG should be distinguished, as disregarding such information leads to the worse performance.
- ATBRG without the representation activation layer performs worse consistently (*i.e.*, ATBRG > ATBRG_{w/o RAL}). It indicates that capturing the semantic relations among user behaviors over the KG can better understand user underlying preference, which is beneficial for the final prediction.

5.3.2 Effect of Neighbor Depth. The proposed ATBRG model is flexible to capture high-order structural information through recursively aggregating the embeddings from an entity's neighbors to refine the entity's embedding in KG. Here, we investigate how the

neighbor depth over KG influences the model performance. Specifically, the neighbor depth of item is explored in the range of {0, 1, 2, 3} in Taobao dataset and {0, 1} for Yelp dataset. We summarize the results in Table 5 and have the following two observations:

- Overall, the model performance increases gradually when neighbor depth varies from 0 to 2 on both datasets. It demonstrates that deepening the neighbor layer helps capture the long-term structural relations of user behaviors and target item to some extent.
- The model performance of ATBRG degrades when the neighbor depth increases from 2 to 3. One possible reason is that long-term relations may include much more ineffective connectives (*i.e.*, *Shirt - Women Clothing - Clothing - Men Clothing - Shoe*). Such relations in the graph introduce some noise and further harm the model performance.

5.3.3 Effect of Aggregator. In our model, we enrich the information of items by recursively capturing their neighbor information in KG. In order to explore the effect of different neighbor aggregator, we design different variants of ATBRG, listed as follows:

- **ATBRG_{concat}**: It applies concatenation operation [22].
- **ATBRG_{sum}**: It applies sum pooling.
- **ATBRG_{sa}**: It applies self-attention mechanism [19].
- **ATBRG_{nl}**: It applies nonlinear transformation [22].

We present the AUC comparison of ATBRG and its variants in Table 6. From the result, we have the following findings:

- Obviously, ATBRG with simple aggregators (*i.e.*, ATBRG_{concat} and ATBRG_{sum}) performs worst on both datasets, since they ignore the different contributions of neighbors.
- Generally, ATBRG with complex aggregators (*i.e.*, ATBRG_{sa} and ATBRG_{nl}) achieves better performance on both datasets. The reason is that ATBRG_{sa} employs self-attention mechanism to place different importance on neighbors while ATBRG_{nl} leverages nonlinear transformation to characterize complex interaction.
- ATBRG consistently yields the best performance on both datasets. It illustrates that, our proposed relation-aware aggregator not only includes the nonlinear transformation in the weight calculation, but also considers the influence of relations during aggregation.

5.4 Case Study (RQ3)

To better understand the merits of our proposed ATBRG intuitively, we first make comprehensive instance-level analyses on the adaptive target-behavior relational graph. As shown in Fig. 3, we present the influence of the number of nodes over the relational graph on the click through rate (CTR). Here, CTR is calculated by averaging the real labels (1 for click and 0 otherwise). It can be clearly observed that CTR and the number of nodes are positively correlated on both datasets. This demonstrates that, the richer the relations between user behavior and target item on the extracted sub-graph, the more likely the user is to click on target item.

Moreover, with the aims of answering how ATBRG addresses the limitations (described in Section 1) existed in previous GNN based methods for knowledge-aware recommendation, we conduct

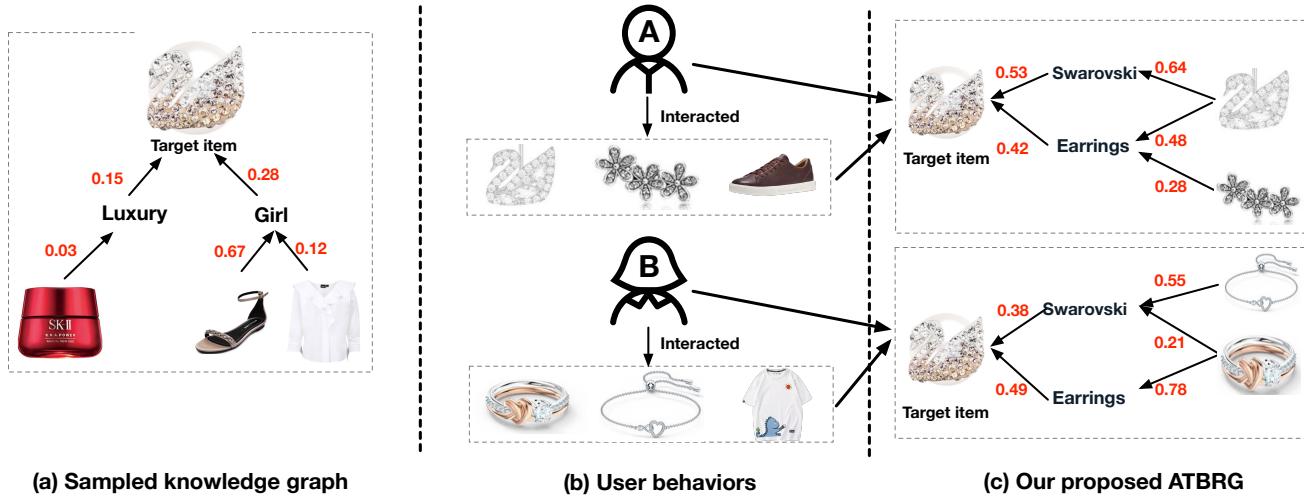


Figure 4: An illustrative example of how our proposed ATBRG works more effective than other knowledge graph based methods. (a) indicates the sampled neighbors of the target item over the knowledge graph. (b) & (c) introduce the specific user behaviors of users and the corresponding extracted sub-graph. The red number above the edge implies the calculated weights.

one case study in large-scale industrial Taobao dataset. As shown in Fig. 4, the main findings are summarized as follows:

- In part (a), due to the limitation (L2), the original neighbors of the target item over the knowledge graph are randomly sampled beforehand. Hence, some relevant entities (*i.e.*, *Swarovski* and *Earring*) connecting users are discarded, while other ineffective entities (*i.e.*, *Luxury* and *Girl*) are reserved, which inevitably introduce noises. It demonstrates that previous methods are incapable of adaptively sampling neighbors for target user-item pairs, and further harm the recommendation performance.
- In part (b), we present the users' recent behaviors, where some behaviors (*i.e.*, *Necklace* and *Earring*) are related to target item over the knowledge graph while others (*i.e.*, *Shoe* and *Skirt*) are not. By the graph connect and prune techniques, we adaptively preserve the effective entities and relations over knowledge graph (L2). Subsequently, in part (c), we construct the specific adaptive target-behavior relational graph for the given target user-item pair, which provides strong evidences for inferring user preference. Propagating embeddings on such a relational structure can take full advantage of the mutual effect of target user-item pair for recommendation (L1).
- In order to consider the rich relations among user behaviors over KG, we propose the relation-aware extractor layer to weigh various underlying preferences for recommendation. Compared with part (a), we find the weights are also adaptive for different users (L3). Specifically, *Swarovski* is paid more attention to and scored higher than *Earring* by the user A, while it is the opposite for user B. Therefore, the final relational representation can reflect the personalized preferences of different users towards the target item.

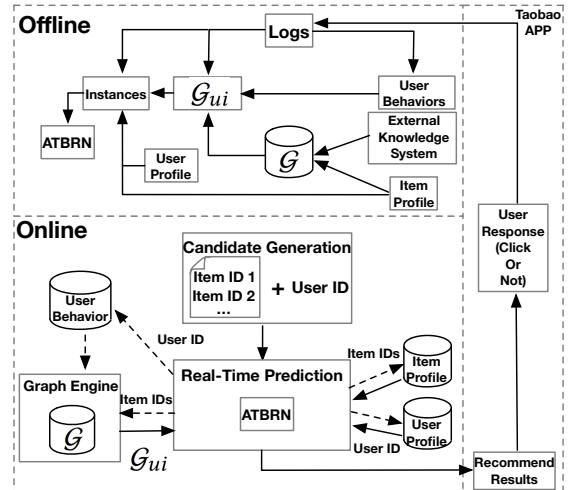


Figure 5: The deployment of ATBRG in Taobao APP.

5.5 Online A/B Testing

To verify the effectiveness of our proposed framework ATBRG in the real-world settings, ATBRG has been deployed in the popular recommendation scenario of Taobao APP. As shown in Fig. 5, the deployment pipelines consist of three parts: 1) User response. Users give implicit feedback (click or not) to the recommended items provided by the recommender system; 2) Offline training. In this procedure, we integrate the knowledge graph \mathcal{G} , user behaviors and target item to construct the adaptive target-behavior relational graph \mathcal{G}_{ui} . Afterwards, \mathcal{G}_{ui} , together with user profile and item profile makes up the instances, and are fed into ATBRG for training; 3) Online serving. When the user accesses Taobao APP, some candidates items are generated by the pipelines before real-time

prediction (RTP) service. The necessary components of ATBRG are achieved and organized in the same way as the offline training. At last, the candidates are ranked by the predicting scores of ATBRG, and truncated for the final recommend results.

Compared with existed deployed baseline model DIN, **6.8%** lift on click count and **5.1%** lift on CTR are observed for ATBRG, with the cost of **8** milliseconds for online inference. The promotion of recommendation performance verifies the effectiveness of our proposed framework ATBRG.

6 CONCLUSION

In this paper, we propose a novel framework ATBRG for knowledge aware recommendation. To effectively characterize the structure relations over KG, we propose the graph connect and graph prune techniques to construct adaptive target-behavior relational graph. Furthermore, we elaborate on the model design of ATBRG, equipped with relation-aware extractor layer and representation activation layer, which aims to take full advantage of structural connective knowledge for recommendation. Extensive experiments on both industrial and benchmark datasets demonstrate the effectiveness of our framework compared to several state-of-the-art methods. Moreover, ATBRG has also achieved 5.1% improvement on CTR metric in online experiments after successful deployment in one popular recommendation scenario of Taobao APP. In the future, we will consider applying causal inference in KG to improve the interpretability of recommender system.

REFERENCES

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
- [2] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *WWW*. 151–161.
- [3] Covington, Paul, Adams, Jay, Sargin, and Emre. 2016. Deep neural networks for youtube recommendations. In *RecSys*. 191–198.
- [4] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *WWW*. 271–280.
- [5] 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 *IJCAI*. 2301–2307.
- [6] Liyu Gong and Qiang Cheng. 2019. Exploiting edge features for graph neural networks. In *CVPR*. 9211–9219.
- [7] Huirong Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *IJCAI*. 1725–1731.
- [8] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *SIGKDD*. 1531–1540.
- [9] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR*. 505–514.
- [10] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- [11] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
- [12] 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 *SIGIR*. 961–964.
- [13] Xusheng Luo, Luxin Liu, Yonghua Yang, Le Bo, Yuanpeng Cao, Jinhang Wu, Qiang Li, Keping Yang, and Kenny Q. Zhu. 2020. AliCoCo: Alibaba E-commerce Cognitive Concept Net. In *SIGMOD*.
- [14] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. 2019. SDM: Sequential deep matching model for online large-scale recommender system. In *CIKM*. 2635–2643.
- [15] 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 *SIGKDD*. 2671–2679.
- [16] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, and Kun Gai. 2019. Lifelong Sequential Modeling with Personalized Memorization for User Response Prediction. In *SIGIR*.
- [17] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.
- [18] Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering* 27, 2 (2014), 443–460.
- [19] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
- [20] 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 *CIKM*. 417–426.
- [21] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *WWW*. 1835–1844.
- [22] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *WWW*. 3307–3313.
- [23] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *SIGKDD*. 950–958.
- [24] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*. 5329–5336.
- [25] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*. 5329–5336.
- [26] Yikun Xian, Zuohui Fu, S Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *SIGIR*. 285–294.
- [27] Fengli Xu, Jianxian Lian, Zhenyu Han, Yong Li, Yujian Xu, and Xing Xie. 2019. Relation-Aware Graph Convolutional Networks for Agent-Initiated Social E-Commerce Recommendation. In *CIKM*. 529–538.
- [28] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*. 353–362.
- [29] Wayne Xin Zhao, Gaole He, Kunlin Yang, Hongjian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. KB4Rec: A Data Set for Linking Knowledge Bases with Recommender Systems. *Data Intelligence* 1, 2 (2019), 121–136.
- [30] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. 2018. ATRank: An attention-based user behavior modeling framework for recommendation. In *AAAI*. 4564–4571.
- [31] 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*. 5941–5948.
- [32] 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 *SIGKDD*. 1059–1068.
- [33] Xiangmin Zhou, Dong Qin, Lei Chen, and Yanchun Zhang. 2019. Real-time context-aware social media recommendation. *The VLDB Journal* 28, 2 (2019), 197–219.