Behavioral Targeting using Heterogeneous Information Networks for Social Media Advertising

[Scalable Data Science]

Jingru Yang[†], Xiaoman Zhao[†], Ju Fan[†], Gong Chen[§], Chong Peng[§], Sheng Yao[§], Xiaoyong Du[†]

† *Renmin University of China, Beijing, China* § *Tencent Inc, Shenzhen, China*

{hinsover,xiaomanzhao,fanj,duyong}@ruc.edu.cn; {natchen,michaelpeng,windyao}@tencent.com

ABSTRACT

Behavioral targeting plays an important role in social media advertising for capturing users' preferences of ads. While the existing studies of behavioral targeting mainly focus on the user behaviors that have explicit correlations with ads, such as ad clicking and web search, there are many implicit relationships between users and ads on social media platforms, which can be utilized to enhance the prediction of users' preferences of ads. In this paper, we model the implicit relationships between users and ads as a heterogeneous information network (HIN) and study the problem of HIN-enhanced behavioral targeting. We propose a framework that first performs representation learning in the HIN and then uses the learned representations to train a prediction model for behavioral targeting. This paper focuses on studying the *incompleteness* challenge of HIN that results in inferior performance for prediction. We introduce a human-in-the-loop HIN completion approach that judiciously selects the most "beneficial" tasks that ask human for completing the HIN. We conduct extensive experiments on real datasets collected from WeChat, the largest social media platform in China. The experimental results show that our approach is effective at constructing a high-quality HIN at a low cost of human involvement, and the HIN can significantly improve the performance of behavioral targeting.

PVLDB Reference Format:
Authors. .. *PVLDB*, 12(xxx): xxxx-yyyy, 2020.
DOI: https://doi.org/10.14778/xxxxxxx.xxxxxxx

1. INTRODUCTION

Social media advertising has emerged as the most prevalent and effective advertising method [6], and been widely adopted by major social platform markets like Facebook, Twitter and WeChat. For example, WeChat, one of the largest social platforms in China, utilizes social advertising in its Moments to present ads to *one billion* monthly active users[1]. An essential problem in social media advertising is to capture users' preferences of ads, so as to deliver the right ads to the right users in the right context.

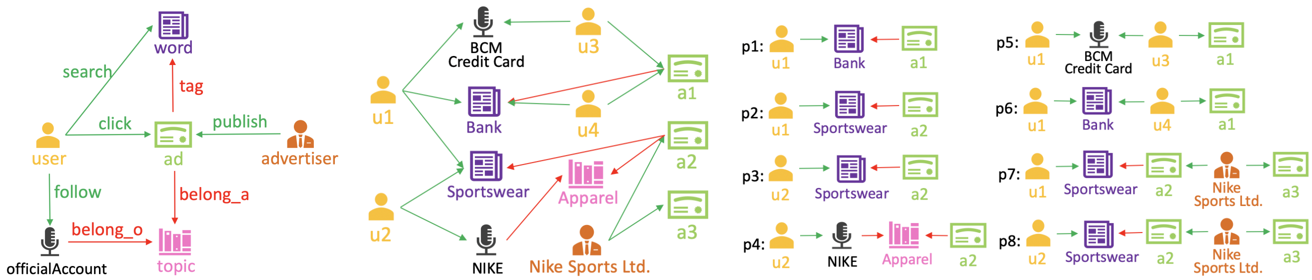
Behavioral targeting (BT) is an effective technique to solve the problem [28, 3]. The basic idea of BT is to collect users' online behaviors, such as web search, and segment the users into different groups. Users in the same group are considered to have similar preferences of ads, and are thus delivered with similar ads. For example, users that frequently search "pasta" and "pizza" can be segmented into one same group and be delivered with similar ads of "pasta" or "pizza". However, in many cases, such user behaviors may not accurately reflect their intent. In our example, a user who searches "pasta" may either want to download recipe Apps or be interested in Italian restaurants. And yet, the existing approaches of BT have limitations on solving the problem, as they only focus on few types of user behaviors with explicit correlation with ads [28, 3, 15, 5, 7], such as ad clicks and web search. In fact, on real social media platforms like WeChat, there are many implicit relationships between users and ads, which can be utilized to enhance the prediction of users' preferences of ads. Consider again the previous example. Suppose that the user recently follows many official accounts related to cooking. This would increase the likelihood that the user prefers ads of recipe Apps.

To address the problem, we propose to utilize the *implicit* relationships between users and ads from a variety of *heterogeneous* sources. We introduce heterogeneous information network (HIN) to capture these implicit relationships. Due to its flexibility and expressiveness, HIN has been proposed as a promising approach to modeling heterogeneous data [20, 21]. Figure 1(b) shows an HIN example with various relationships between users and ads on the WeChat platform, such as official accounts, words, etc. These relationships can be utilized to capture users' preferences of ads.

This paper studies the research challenges that naturally arise in utilizing the HIN for behavioral targeting in real advertising platform in WeChat. First, construction of a high-quality HIN in the advertising scenario is very challenging, as it is inherently a *human-in-the-loop* process. Specifically, although some types of nodes and edges can be obtained by automatic processes, many others are *incomplete* and need be provided by human. For example, the tags of an ad are usually provided by the advertiser, depending on which audience the advertiser wants to reach. Moreover, it is very common in practice that much information of ads, such as

[1]http://3g.163.com/tech/article/E0JHNVMK00097U7R.html

Proceedings of the VLDB Endowment, Vol. 12, No. xxx
ISSN 2150-8097.
DOI: https://doi.org/10.14778/xxxxxxx.xxxxxxx



(a) ADS-HIN Schema. (b) ADS-HIN Example. (c) Paths Example.

Figure 1: An Example of HIN in Behavioral Targeting.

brand and category, is missing. In these cases, the platform needs to solicit the workers in its internal crowdsourcing platform for completing the information. The key challenge here is that it is prohibitive to ask human to do exhaustive completion. To address the challenge, we propose a method to assign the most "beneficial" tasks to human for verification under a budget. We propose a *utility* function that determines what constitutes a "beneficial" completion tasks, and devise an active HIN completion algorithm based on the utility. The second challenge is how to use the HIN to enhance behavioral targeting. We devise a graph representation model to encode users and ads into low-dimensional embeddings, and use the embeddings to train a prediction model for behavioral targeting.

The contributions of this paper are summarized as follows.

(1) To the best of our knowledge, we are the first to study HIN enhanced behavioral targeting on *large-scale* and *real* social media advertising datasets collected from WeChat, the largest social platform in China.

(2) We focus on solving the incompleteness challenge of HIN. We introduce a utility function that determines what constitutes a "beneficial" completion task, and device an active HIN completion algorithm based on the utility.

(3) We conduct an extensive experimental study, and the experimental results show that our approach is effective at constructing a high-quality HIN at low cost of human involvement, and the HIN can significantly improve the performance of behavioral targeting.

2. PROBLEM FORMULATION

In this paper, we study the behavioral targeting problem based on a heterogeneous information network (HIN) with the support of various types of information sources in advertising. More formally, let $U = \{u\}$ denote a set of users and $A = \{a\}$ denote a set of advertisements (or ads for simplicity). We define an HIN in advertising as follows.

DEFINITION 1 (HIN IN ADVERTISING (ADS-HIN)). *The heterogeneous information network in advertising (Ads-HIN) is defined as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a node set $\mathcal{V}$, an edge set $\mathcal{E}$ and two type mapping functions: 1) a node type function $\phi : \mathcal{V} \to \mathcal{V}_{\text{type}}$ that maps a node v to a node type in $\mathcal{V}_{\text{type}}$, and 2) an edge type function $\psi : \mathcal{E} \to \mathcal{E}_{\text{type}}$ that maps an edge e to an edge type in $\mathcal{E}_{\text{type}}$. In particular, we consider users U and ads A must be included as nodes, i.e., $U \subset \mathcal{V}$ and $A \subset \mathcal{V}$. Schema of the Ads-HIN is defined as a directed graph $\mathcal{G}_S = (\mathcal{V}_{\text{type}}, \mathcal{E}_{\text{type}})$ where $\mathcal{V}_{\text{type}}$ and $\mathcal{E}_{\text{type}}$ are the sets of node types edge and types respectively.*

Figure 1(a) shows the schema of an ADS-HIN we built on the data from WeChat. It includes 6 node types in $\mathcal{V}_{\text{type}}$, namely **`user`**, **`ad`**, **`advertiser`**, **`officalAccount`**, **`word`**, and **`topic`**, and 7 edge types in $\mathcal{E}_{\text{type}}$, namely **`click`**, **`follow`**, **`search`**, **`publish`**, **`tag`**, **`belong_a`** and **`belong_o`**. For example, the edge type **`click`** from node type **`user`** to **`ad`** represents the clicking behavior of users on ads, and the edge type **`follow`** from **`user`** to **`officialAccount`** indicates the following behavior of users on official accounts in WeChat. Based on the schema, we build the following ADS-HIN.

EXAMPLE 1. *Figure 1(b) shows an example of* ADS-HIN *that consists of nodes and edges with the types defined in the schema. We can see that the* ADS-HIN *contains information from different views can be used to capture the potential interest of users on ads, which is illustrated as the paths in Figure 1(c). For example, the information from keywords that a user searches (e.g., paths $p_1 - p_3$) indicates that the user may be interested in the ads with the keywords as their tags. The information from the official accounts that a user follows (e.g., p_4) indicates that the user may be interested in the ads in similar topics with the official accounts. The information from the behavior of other users that follow the same official account with a user (path p_5) may be useful to infer the user tends to have similar behaviors. In addition, more complicated relationships between users and ads, e.g., paths $p_6 - p_8$, can also be exploited.*

Given the ADS-HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we formalize the behavioral targeting problem in advertising as learning a classification function $f : U \times A \to \{0, 1\}$, where various kinds of heterogeneous information in $\mathcal{G}$ are incorporated to learn the function. More formally, let $\boldsymbol{x}_u$ denote the *intrinsic* features of user u, such as education and age, and $\boldsymbol{x}_a$ denote the intrinsic features of ad a, such as media format and size. Moreover, we use $\boldsymbol{e}_u$ and $\boldsymbol{e}_a$ to represent the learned features from our ADS-HIN $\mathcal{G}$ (more details of how to learn these features will be described later). Then, the problem of HIN enhanced behavioral targeting is defined as follows.

DEFINITION 2 (HIN ENHANCED BT). *Consider a set of training examples $\mathcal{T} = \{(\boldsymbol{x}_u, \boldsymbol{x}_a; y)\}$ for a behavioral targeting task, where $\boldsymbol{x}_u$ and $\boldsymbol{x}_a$ are receptively intrinsic features of user u and ad a and $y \in \{0, 1\}$ is the targeting result. Consider an* ADS-HIN *$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the learned features $\boldsymbol{e}_u$ and $\boldsymbol{e}_a$ for u and a. It aims to learn a classification function $\hat{y} = f(\boldsymbol{x}_u, \boldsymbol{x}_a; \boldsymbol{e}_u, \boldsymbol{e}_a)$ that maps features of a user-ad pair (u, a) into a behavioral targeting result $\hat{y}$.*

To illustrate the problem defined above, we use click-through rate (CTR) prediction [8], which is a common evaluation for behavioral targeting as an example.

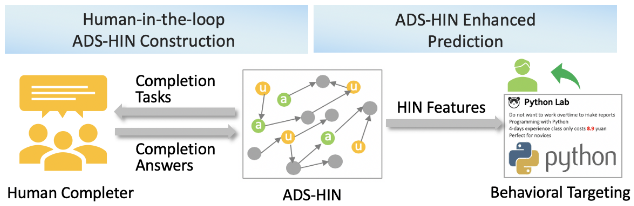


Figure 2: The Framework of Our Approach.

Example 2. *CTR prediction is to predict if a user will click an ad. CTR is an important factor in behavioral targeting, as the advertising platform normally uses* $CTR \times bid$ *to estimate the total revenue, where bid is the benefit the platform receives if an ad is clicked by a user. To address this task, we use a set* $\mathcal{T}$ *of historical clicking behaviors of users on ads as training examples, and with the help of our* Ads-HIN*, we train a classification function* $\hat{y} = f(\boldsymbol{x}_u, \boldsymbol{x}_a; \boldsymbol{e}_u, \boldsymbol{e}_a)$ *to predict whether a user* u *will click ad* a *in the future.*

3. OUR APPROACH

To support HIN enhanced behavioral targeting, we develop an approach to learn "high-quality" predictive features from the Ads-HIN.

3.1 Framework of Our Approach

Our framework, as shown in Figure 2, takes as input a training set $\mathcal{T}$ for specific task for BT (e.g., CTR prediction), and outputs a prediction model using two steps.

Step 1: Human-in-the-loop Ads-HIN construction. This step aims at constructing a high-quality Ads-HIN based on a predefined schema, as shown in Figure 1(b).

However, it is very challenging to construct the Ads-HIN solely using *automatic* processes. Take our example schema shown in Figure 1(a) as an example. While edges with some specific types (in green color), such as `search` and `follow`, can be automatically created by collecting users' and advertisers' behaviors in the advertising platform, edges with other types (in red color) are not easy to be obtained. For example, in a real advertising platform, the tags of an ad are usually provided by the advertiser, depending on which audience the advertiser wants to reach. Moreover, as ads and official accounts may contain contents in heterogeneous formats, such as text, image and video, the existing topic detection algorithms may not achieve superior performance.

To address this problem, this paper introduces a *human-in-the-loop* method to construct the Ads-HIN. It first partially constructs an Ads-HIN based on the data that can be automatically collected in the advertising platform (i.e., edges in green color). Then, it solicits human to "complete" the edges that are missing in the Ads-HIN (i.e., edges in red color). In our practice in the real-world advertising platform of WeChat, we have various ways to leverage human for this completion task, e.g., requesting advertisers to provide tags for their ads, asking data labeling employees for the completion task answers and publishing these tasks to the company's internal crowdsourcing platform.

Obviously, it is prohibitive to ask human to do exhaustive completion for our large scale Ads-HIN in real advertising platforms. Thus, our framework assigns the most "beneficial" edges to human for verification under a budget, i.e., an affordable number of edges to be completed. The fundamental challenge here is to determine what constitutes a "beneficial" edge and should therefore be verified by human. To this end, we propose a *utility* function that takes the following two factors into consideration. The first factor is whether an edge could be verified to be true, e.g., whether a tag can be assigned to an ad. We prefer to ask human to verify the edges with high existent possibility, in order to avoid waste of our budget. The second factor is the degree of *informativeness* of an edge, if verified by human, on improving the performance of our prediction model. We will present more details of these two factors in Section 4.

Step 2: Ads-HIN enhanced prediction. This step aims at "extracting" features for users and ads from the constructed Ads-HIN, and feeds the features into the training process of prediction model f. To this end, we devise a graph representation learning method to map each node in Ads-HIN to a low-dimensional vector. Then, we utilize a prediction model that concatenates both intrinsic and Ads-HIN-learned features and outputs the prediction results.

For ease of presentation, we first present the Ads-HIN enhanced prediction model in Section 3.2, assuming that the Ads-HIN has already been constructed. Then, we devise an active completion algorithm that judiciously leverages human to complete the Ads-HIN in Section 4.

3.2 ADS-HIN Enhanced Prediction

This section presents our Ads-HIN enhanced prediction model for behavioral targeting. The basic idea is to first utilize the feature concatenation method, which is simple and widely adopted for feature enhancement [25, 23], to enhance intrinsic features of behavioral targeting with HIN features mined from $\mathcal{G}$. Then, we adopt a binary classification model, such as DeepFM [8] for CTR prediction. Formally, given a user u and an ad a, we consider the following features.

Intrinsic Features. Like the conventional solutions for behavioral targeting, we consider the *intrinsic features* for both user u and ad a, as mentioned previously. To be more specific, we use $\boldsymbol{x}_u$ to describe the user profile of u with multiple fields, such as age, gender, education background, etc. We represent the i-th field of user profile as a one-hot vector $\boldsymbol{x}_u^{(i)}$ and denote $\boldsymbol{x}_u$ as $\boldsymbol{x}_u = \boldsymbol{x}_u^{(1)} \oplus \boldsymbol{x}_u^{(1)} \oplus \ldots \boldsymbol{x}_u^{(m)}$. Similarly, we use $\boldsymbol{x}_a$ to represent features of ad a with multiple fields, such as media format (image or video) and size.

HIN Features. This paper focuses on incorporating *HIN features* mined from our Ads-HIN $\mathcal{G}$ for the behavioral targeting problem. Intuitively, this features capture whether user u and ad a are "close" w.r.t. the graph connectivity in $\mathcal{G}$. For example, if u and a are frequently connected in multiple paths in $\mathcal{G}$, we consider that u are more likely to be interested in a. To formalize such "closeness" between u and a in our Ads-HIN $\mathcal{G}$, we adopt the translation-based embedding model TransE [2], which is widely-adopted for HIN or knowledge graph. The rationality of using the model is that it can transform nodes and edge types in $\mathcal{G}$ into continuous vector space, where nodes that are close in graph will be transformed into similar vectors. Thus, we can easily feed the vectors of user u and ad a, denoted as $\boldsymbol{e}_u$ and $\boldsymbol{e}_a$, as HIN features into our prediction model.

Formally, we also use a triplet $s = (v_h, v_t, r)$ to denote an edge with type r from node v_h to node v_t for ease of presentation, where v_h and v_t are called *head* and *tail* nodes for simplicity. The basic idea of TransE [2] is to consider, in each triplet, head node v_h can be "translated" into tail node

v_t via the specific edge type r, i.e., the learned embeddings of nodes and edge type satisfy $\mathbf{e}_{v_h} + \mathbf{e}_r \approx \mathbf{e}_{v_t}$. To formalize this, TransE introduces an energy score function as,

$$d(s) = ||\mathbf{e}_{v_h} + \mathbf{e}_r - \mathbf{e}_{v_t}||_2, \tag{1}$$

where $||\cdot||_2$ is L_2-norm distance function. Based on this, it defines a margin-based ranking loss function to training the embedding-based representation from our ADS-HIN $\mathcal{G}$, i.e.,

$$\mathcal{L}_{\mathtt{G}} = \sum_{s\in\mathcal{G}} \sum_{s^{'}\in\mathcal{G}'} [\delta + d(s) - d(s^{'})]_+, \tag{2}$$

where $[x]_+ = max\{0, x\}$ and δ is the margin, and $\mathcal{G}^{'}$ is the set of negative triplets that are constructed by replacing head or tail node to a random node.

Finally, we concatenate the intrinsic and HIN features $\boldsymbol{x} = \boldsymbol{x}_u \oplus \boldsymbol{x}_a \oplus \boldsymbol{e}_u \oplus \boldsymbol{e}_a$ to train a prediction model $\hat{y} = f(\boldsymbol{x})$.

4. ACTIVE ADS-HIN COMPLETION

In our framework, we would like to ask human to complete some "missing" edges in our ADS-HIN, which are difficult to determine by machine. Formally, we define the *completion task* performed by human.

DEFINITION 3 (COMPLETION TASK). *A completion task $\omega = (v_h, r, \mathcal{V}_{\mathtt{cand}})$ consists of a head node v_h, an edge type r and a set of candidate tail nodes $\mathcal{V}_{\mathtt{cand}}$. Each candidate $v_t \in \mathcal{V}_{\mathtt{cand}}$ corresponds to a possible edge (v_h, v_t) with edge type r, which is also denoted as a triplet $s = (v_h, v_t, r)$ for ease of presentation. The answer of the task is a subset $\mathcal{V}^* \in \mathcal{V}_{\mathtt{cand}}$ of the candidates that are verified by human.*

For example, in Figure 1(b), suppose that the `belong_o` edge between official account `NIKE` and topic `Apparel` is missing. In this case, we will publish a task that consists of `NIKE` as head node v_h and `belong_o` as edge type r and a set of candidate topics as $\mathcal{V}_{\mathtt{cand}}$. The human is asked to select the correct topics from the candidates for `NIKE`.

We would like to select the most "beneficial" tasks to human for completion under a given budget. To this end, we introduce *task utility* as a criterion for task selection in Section 4.1, and present an algorithm for active ADS-HIN completion based on the utility in Section 4.2.

4.1 Task Utility Estimation

Intuitively, we favor the "informative" tasks, if completed by human, which would be most helpful to enhance our prediction model. To formalize this idea, we first introduce two factors for each triplet $s = (v_h, v_t, r)$. The first factor is *truthfulness* $P(v_t|v_h, r)$, which represents the probability that triplet $s = (v_h, v_t, r)$ can be successfully completed by human. The second factor is *informativeness* $\mathcal{I}(v_h, v_t, r)$, which captures the degree of triplet $s = (v_h, v_t, r)$ on enhancing our prediction model. We will discuss more details of informativeness later. Based on these two factors, the utility of the task ω is defined as the overall degree of informativeness that would be brought by ω, i.e.,

$$\mathcal{U}(\omega) = \sum_{v_t\in\mathcal{V}_{\mathtt{cand}}} P(v_t|v_h, r)\cdot\mathcal{I}(v_h, v_t, r). \tag{3}$$

Next, we present how to estimate the truthfulness and informativeness of an edge as follows.

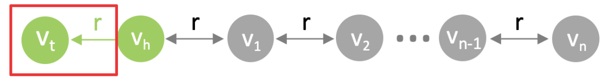


Figure 3: Simple Example Graph.

4.1.1 Estimation of Truthfulness.

Intuitively, the truthfulness $P(v_t|v_h, r)$ of a triplet $s = (v_h, v_t, r)$ captures the probability that s will be verified by human. We use our learned HIN embedding using TransE to estimate $P(v_t|v_h, r)$ (see Section 3.2). The intuition is that, based on TransE, the smaller the translational distance $d(s)$ is, the larger the probability that an edge with type r exists from v_h to v_t, i.e., the more likely that s can be verified to be true. Thus, we adopt translation-based energy function (TEF) [14] on top of $d(s)$ to estimate the truthfulness, i.e.,

$$P(v_t|v_h, r) = \frac{1}{1 + e^{-\alpha(\tau_r - d(s))}}, \tag{4}$$

where α is a hyper-parameter used for smoothing, and τ_r is a threshold related to type r which can be obtained by maximizing classification accuracy on the validation triplets of type r. We can see that, when $d(s) = \tau_r$, the value of $P(v_t|v_h, r)$ is 0.5, and when $d(s) < \tau_r$, $P(v_t|v_h, r) > 0.5$.

4.1.2 Estimation of Informativeness.

Intuitively, the informativeness $\mathcal{I}(s)$ of triplet s captures how helpful s is to enhance our prediction model. Take Figure 1(b) as an example. We consider the completion from a_2 to `Sportswear` may be more informative than that from a_1 to `Bank`. The reason is that the former, if verified to be true, will connect two disconnected nodes u_1 and a_2. TransE model will transform nodes u_1 and a_2 into more "close" vectors in the embedding space, which would provide additional evidence to our prediction model that u_1 may be interested in a_2.

From the example, we can see that we prefer the triplets, which, if added into our ADS-HIN $\mathcal{G}$, would have large effect on updating the learned embeddings of users and ads, i.e., $\{\boldsymbol{e}_u\}$ and $\{\boldsymbol{e}_a\}$, which would result in the translational distance between u and a to be significantly changed. As our prediction model takes the embeddings as its input, these triplets essentially give us more chances to update our prediction model f for behavioral targeting. Note that this is conceptually similar to a widely-adopted criterion in active learning [33, 17] that prefers the data points having larger effect on updating the model.

Formally, let us consider the set of user-ad pairs in our training set $\mathcal{T} = \{(u, a)\}$ and our ADS-HIN $\mathcal{G}$. We use $d_{\mathcal{G}}(u, a, r_c)$ to represent the translational distance between u and a given the edge type r_c corresponding to the behavioral targeting task. For example, considering the CTR prediction task, r_c corresponds to the edge type `click`. Moreover, let $d_{\mathcal{G}\cup\{s\}}(v_u, v_a, r_c)$ denotes the translational distance of (u, a) after a triplet s is added to $\mathcal{G}$. We define the informativeness $\mathcal{I}(s)$ of triplet s as *changes* on the translational distances among all user-ad pairs in our training set $\mathcal{T}$, i.e.,

$$\mathcal{I}(s) = \sum_{(u,a)\in\mathcal{T}} (d_{\mathcal{G}}(v_u, v_a, r_c) - d_{\mathcal{G}\cup\{s\}}(v_u, v_a, r_c))^2. \tag{5}$$

The non-trivial part in Equation (5) is the computation of the translational distance $d_{\mathcal{G}\cup\{s\}}(v_u, v_a, r_c)$ in the updated $\mathcal{G}\cup\{s\}$. A straightforward way is to perform graph embedding learning, which is described in Section 3.2, in $\mathcal{G}\cup\{s\}$.

However, this method is very prohibitive, as it needs to rerun graph embedding learning for each possible triplet s.

To address the problem, we propose an effective approach to estimate $d_{\mathcal{G}\cup\{s\}}(v_u, v_a, r_c)$ instead of rerunning the embedding learning process. The basic idea of the approach is to estimate the updated embedding $\boldsymbol{e}_v$ for each node v in the updated $\mathcal{G}\cup\{s\}$, based on the definition of $d_{\mathcal{G}\cup\{s\}}(v_u, v_a, r_c)^2$. For ease of presentation, we use a simple graph as shown in Figure 3 to describe our estimation method, and we will discuss the general graph later.

Let (v_h, v_t, r) be a triplet that is newly added into the graph. The embeddings of each node v_n in the graph will be updated as the gradient information of (v_h, v_t, r) is propagated through the interlink in Figure 3. For simplicity of the estimation, we approximate the propagation as the first iteration update of the embedding of v_n caused by (v_h, v_t, r), which is the similar approximate method of the existing active learning work [9]. Formally, let $\mathbf{e}^+_{v_n}$ denotes the new embedding of v_n after the completion of triplet (v_h, v_t, r).

Lemma 1. *Consider the simple graph in Figure 3 and a newly added triplet (v_h, v_t, r). The updated embedding $\mathbf{e}^+_{v_n}$ can be approximated as*

$$\boldsymbol{e}^+_{v_n} \leftarrow \boldsymbol{e}_{v_n} - (2\gamma)^{n+1}(\boldsymbol{e}_{v_h} + \boldsymbol{e}_r - \boldsymbol{e}_{v_t}), \tag{6}$$

where n is the length of interlink connecting v_n and v_h, γ is the learning rate for back propagation.

Due to the space limit, we put the proof of Lemma 1 in our technical report [29]. We can extend Lemma 1 to the more general case that there are multiple paths between a node v_n and the newly completed triplet $s = (v_h, v_t, r)$, i.e.,

$$\mathbf{e}^+_v = \mathbf{e}_v - (\mathbf{e}_{v_h} + \mathbf{e}_r - \mathbf{e}_{v_t}) \Bigg[\sum_{p_v^{v_h} \in P_v^{v_h}} (2\gamma)^{|p_v^{v_h}|+1} - \sum_{p_v^{v_t} \in P_v^{v_t}} (2\gamma)^{|p_v^{v_t}|+1} \Bigg], \tag{7}$$

where $P_v^{v_h}$ is the set of paths between v_h and v, and $P_v^{v_t}$ is the set of paths between v_t and v. For simplicity, we use ρ_v to denote $\sum_{p_v^{v_h} \in P_v^{v_h}} (2\gamma)^{|p_v^{v_h}|+1} - \sum_{p_v^{v_t} \in P_v^{v_t}} (2\gamma)^{|p_v^{v_t}|+1}$ of node v. Now we are ready to obtain the informativeness $\mathcal{I}s$ of triplet s by applying Equation (7) in Equation (5), i.e,

$$\begin{aligned} \mathcal{I}(s) &= \sum_{(u,a)\in\mathcal{T}} \Big(||\mathbf{e}_{v_u} + \mathbf{e}_{r_c} - \mathbf{e}_{v_a}||^2 - ||\mathbf{e}^+_{v_u} + \mathbf{e}_{r_c} - \mathbf{e}^+_{v_a}||^2 \Big)^2 \\ &= \sum_{(u,a)\in T} \Big(2(\mathbf{e}_{v_u} + \mathbf{e}_{r_c} - \mathbf{e}_{v_a})(\mathbf{e}_{v_h} + \mathbf{e}_r - \mathbf{e}_{v_t})(\rho_{v_u} - \rho_a) \\ &\qquad - (\mathbf{e}_{v_h} + \mathbf{e}_r - \mathbf{e}_{v_t})^2 (\rho_{v_u} - \rho_{v_a})^2 \Big)^2 \end{aligned} \tag{8}$$

4.2 Utility-based Completion Algorithm

The pseudo-code of our utility-based completion algorithm is shown in Algorithm 1. It takes as input a current Ads-HIN $\mathcal{G}$, a set Ω of completion tasks, a budget constraint B on the number of tasks and a batch size b. The output of the algorithm is an updated $\mathcal{G}$ completed by human. Initially, the algorithm first trains a TransE model [2] on $\mathcal{G}$ (line 1). Then it repeats a task selection iteration until the budget B is exhausted or there is no remaining tasks (lines 2-10). In each iteration, it first estimates the utility $\mathcal{U}(\omega)$ of each task ω in Ω based on Equation (3) (line 3). Then, it delivers b tasks with the highest utility to human, collects the answers and adds the newly completed triplets into $\mathcal{G}$ (lines 4-6). Finally, it retrains the TransE model on the updated $\mathcal{G}$ and continues to the next iteration.

One obstacle in the algorithm is the efficiency of retraining the TransE model (line 7). To address the obstacle, we introduce the following two strategies to improve the training efficiency while ensuring the training quality. The first strategy is to update embeddings of a subgraph of $\mathcal{G}$, instead of the entire $\mathcal{G}$. Observing from the training result of TransE, we find that the effect of a new triplet s on the embedding of a node decreases exponentially with the increase of the length of the path from s to the node. Based on this observation, after adding a new triplet s, we can use TransE model to only update the embeddings in the subgraph containing nodes within a distance l from s, where l is a hyper-parameter that can be tuned in the experiments.

The second strategy is to train the embeddings in the subgraph of each newly added triplet in parallel. To this end, we assign the subgraphs of the newly added triplets to different executors for parallel training. In particular, for a node or edge type that is assigned to multiple executors, we input the embeddings obtained from these executors into a max-pooling layer to obtain a unified embedding, for keeping the most prominent features [16].

[2]Since the update of embedding $\boldsymbol{e}_r$ of edge type r is the addition of the update of its corresponding head and tail nodes, in order to simplify the expression, we consider $\boldsymbol{e}_r$ remains unchanged in the updated graph.

Algorithm 1: Active Ads-HIN Completion

Input: $\mathcal{G}$: current Ads-HIN; Ω: completion takes; B: a budget; b: a task batch
Output: $\mathcal{G}$: completed Ads-HIN

```
1  Train a TransE model on G;
2  for each iteration do
3      Estimate U(ω) for each task ω in Ω;
4      Select b tasks Ω* from Ω with the largest U(ω);
5      Ask the human to complete the tasks in Ω* ;
6      Add the triplets completed by human in G ;
7      Retrain the TransE model on G ;
8      B ← B − b ;
9      Ω ← Ω − Ω* ;
10     if B = 0 or Ω = ∅ then break;
11 Return G ;
```

Table 1: Statistics of Ads-HIN in WeChat platform

# of nodes with various types			
user	1,167,481	ad	541,118
advertiser	43,188	official_account	255,221
word	243,048	category	236
brand	1,929	tag	256
# of edges with various types			
click	554,691	interact	47,425
un-interested	248,850	advertise	525,422
follow	4,101,158	interested_in_tag	999,200
interested_in_word	998,882	mention	2,737,389
belong_to_topic	792,377	belong_to_brand	1,192,886
advertising_to_tag	1,371,282		

5. EXPERIMENTS

5.1 Experiment Setup

Dataset. We construct a large-scale dataset based on data collected from the real social advertising platform in WeChat.

The dataset consists of an Ads-HIN and the training sets for two representative behavioral targeting tasks.

First, we build an Ads-HIN $\mathcal{G}$ based on the information from different views in WeChat to capture the potential interest of users on ads. More specifically, we define 8 types of nodes and 11 types of edges in the schema of $\mathcal{G}$. Among the edge types, we consider four types, namely `mention`, `belong_to_topic`, `belong_to_brand` and `advertising_to_tag`, need to be completed by human, as the automatic process cannot provide accurate results for these types. Edges with other types, such as `click` and `follow`, can be automatically constructed based on the data in WeChat. The statistics of the Ads-HIN are shown in Table 1, while more details of the schema are included in our technical report [29].

Second, we construct datasets for two representative behavioral targeting tasks in social advertising. The first task is *ad matching* that measures whether a user is interested in an ad, and the second one is *CTR prediction* that predicts whether a user will click the ad that is delivered to the user. For each task, we prepare a dataset based on the real data log from WeChat, where each data instance consists of the intrinsic features of users and ads, which are presented previously, and an output for prediction, e.g., matched or clicked. We take the log data of one week as the training set, and take the data of the day next to the week as the test set. Based on this, we obtain a dataset with $5,380,531$ data instances for CTR prediction where the ratio between positive and negative instances is $1:5$, and a data set for ad matching with positive/negative ratio $1:500$.

Prediction Models. To evaluate the effect of the Ads-HIN as auxiliary data to enhance ad matching and CTR prediction tasks, we concatenate the HIN features and the intrinsic features and use the following state-of-the-art predict models for evaluation. (1) DSSM [11] is one of the most widely used ad matching models, which uses two deep neural networks to convert the one-hot encoded user features and ad features into low-dimensional vectors separately. (2) CDSSM [18] is a variant of DSSM, which incorporates a convolutional-pooling structure over user features and ad features to learn their low-dimensional vectors. (3) DeepFM [8] is one of the most widely used CTR prediction model, which combines the power of factorization machines and deep learning. (4) Fibinet [12] is a state-of-the-art CTR prediction model, which combines feature importance and bilinear feature interaction.

Comparison Method for Ads-HIN Completion. To evaluate the effectiveness of our active completion algorithm presented in Section 4.2, we consider the following baseline methods for selecting completion tasks. (1) *Random* randomly selects tasks for completion. (2) *Uncertain* prefers to select tasks with uncertain triplets. Specifically, a triplet is uncertain if its truthfulness is close to 0.5. Formally, we use the information entropy to measure the uncertainty. (3) *Link-Only* utilizes the reciprocal of the length of the shortest path between each triplet and each user-ad pair in the Ads-HIN. (4) *Trust-Only* is a variant of our triplet utility score in Equation (3) that only considers the truthfulness while ignoring the informativeness. (5) *Info-Only* is a variant of our triplet utility score in Equation (3) that only considers the informativeness while ignoring the truthfulness.

Moreover, to demonstrate efficiency and effectiveness of the incremental training strategy introduced in Section 4.2, we consider the following baseline training methods. (1) `Retrain` adds newly completed triplets into our Ads-HIN and retrains TransE model from scratch. (2) `NewData` only considers the newly completed triplets to fine-tune TransE model, while keeping embeddings of other nodes unchanged.

Table 2: Evaluating HIN Feature Enhancement for Ad Matching and CTR Prediction Tasks.

Method	Ad Matching				CTR Prediction			
	HR@0.1%		HR@5%		AUC		LogLoss	
	DSSM	CDSSM	DSSM	CDSSM	DeepFM	Fibinet	DeepFM	Fibinet
`NonHIN`	0.016	0.017	0.262	0.247	0.692	0.694	0.437	0.435
`PartHIN`	0.064	0.068	0.435	0.412	0.706	0.709	0.434	0.432
`CompHIN`	0.116	0.117	0.557	0.550	0.727	0.728	0.423	0.423

Evaluation metrics. For the ad matching task, we use Hit Ratio (HR) as the evaluation metric, which measures the ratio of matched ads in prediction results returned by the prediction model. For the CTR prediction task, we measure the performance using the following two metrics: 1) area under the ROC curve, i.e., Area Under Curve (AUC), and 2) binary cross entropy loss (LogLoss), which are widely used in the existing works for CTR [8, 12].

Parameter settings. For triplet completion, We pick up 10,000 triplet tasks at each iteration. For TransE model training, we set the dimension of the embeddings as 100 and margin as 1.0. For incremental graph training, at each iteration of the HIN update, we set the subgraph path threshold l as 2 and set 4 executors for the parallel training. For the prediction models, we use the hyper-parameters which are reported effective in their original papers. More specifically, for DSSM, we use two three-layer neural networks whose number of units in the first layer is 300 and the number of units in other layers is 128. For CDSSM, on the basis of DSSM, we use 32 filters in the convolution and a kernel size of 3. For DeepFM, we use a two-layer neural network with 64 hidden units and an embedding size of 32. For Fibinet, we use a three-layer neural network with 400 hidden units and an embedding size of 50.

5.2 Experimental Results

This section presents the experimental results on the two behavioral tasks, ad matching and CTR prediction.

Evaluation on HIN Features. We first evaluate the effect of HIN features mined from our Ads-HIN on enhancing the prediction model. To this end, we consider three settings of the features that are fed into the prediction model. (1) `NonHIN` only feeds the intrinsic features of users and ads. (2) `PartHIN` only considers the *partial* Ads-HIN without human completion, and concatenates the HIN features learned from the graph with the intrinsic features to the prediction model. (3) `CompHIN` is similar to `PartHIN` where the only difference is that the Ads-HIN has been completed by human.

Table 2 shows the experimental results. We can see that the consideration of HIN features can improve the performance of the two tasks significantly. For example, `PartHIN` has 66% improvements on HR@n% and 2.16% improvements on AUC, and `CompHIN` has 625% improvements on HR@n% and 5.06% improvements on AUC. Moreover, for the two tasks, `CompHIN` outperforms `PartHIN`, e.g., 28%-81% improvements on HR@n%, up to 2.97% improvement on AUC, and up to 2.6% reductions on LogLoss. These observations, on the one hand, validate our claim that the HIN features are useful to enhance the prediction model for behavioral targeting, and, on the other hand, show the necessity of human-in-the-loop Ads-HIN completion.

Table 3: **Evaluating efficiency of training methods.**

Training Method	TransE Training Time (min)
`Incremental`	295.53
`NewData`	259.32
`Retrain`	653.28

Evaluation on Active HIN Completion. We evaluate different strategies for HIN completion. For ease of presentation, we denote our utility-based strategy as AcHC. For more comprehensive comparisons, we vary the budget for HIN completion, which is the percentage of tasks published for completion in all candidate tasks.

Figure 4 reports the experimental results. We can see that *Random* performs the worst, which shows that active HIN completion is necessary. The two baselines *Uncertain* and *Trust-Only* also achieve inferior performance, because they only consider whether more triplets can be added in Ads-HIN, while ignoring if the newly added triplets are informative to improve the prediction model. Moreover, *Link-Only* achieves better performance than the above two baselines. This is because *Link-Only* prefers the triplets that are closed to the user-ad pairs in our training set $\mathcal{T}$. Recall our previous analytics that the effect of triplets on the embeddings of a node decreases exponentially with the increase of the length of the paths from the triplets to the node. Due to this reason, *Link-Only* gives more chances for users and ads to update the embeddings after the triplet completion, which may further improve the prediction model. However, *Info-Only* performs better than *Link-Only*, because it considers the distances in translational distances between users and ads, which would further improve the informativeness. Moreover, AcHC further improves *Info-Only*, as it considers both truthfulness and informativeness in measuring the task utility. We can also see that AcHC achieves nearly the results of `CompHIN` with only 20% of the triplet tasks, which are reported in Table 2. These results demonstrate the effectiveness of our completion method for the real large-scale advertising scenario that reduces the effort of preparing the Ads-HIN for behavioral targeting.

Evaluation on Incremental Training. We first compare the efficiency of the three training strategies. Table 3 reports the time of training the TransE model in all the iterations. We can see that `Incremental` significantly reduces the total training time over `Retrain`, because it only considers the path within length threshold of 2 and reduces the training data in TransE by nearly 92% in each iteration of the Ads-HIN updating. On the other hand, although `NewData` can reduce 97% of the training data in each iteration, it does not significantly outperform `Incremental` as the parallel training of `Incremental` further saves the time.

Next, we compare the performance of the three training methods on the CTR prediction and ad matching tasks, as shown in Figure 5. We observe that `NewData` performs the worst, because it fine-tunes the TransE model trained on the previous iterations by only the newly completed triplets, which may lose much information in the graph. Second, `Retrain` uses all the triplets, including the original triplets and the newly completed triplets to retain the graph, and it achieves the best performance. However, the differences in performance between `Incremental` and `Retrain` is insignificant. In particular, as the number of completed triplets increases, the difference becomes even smaller. Overall, the above results show that our proposed `Incremental` method can significantly reduce the training time while effectively preserving the quality of training results.

5.3 Case Study in WeChat

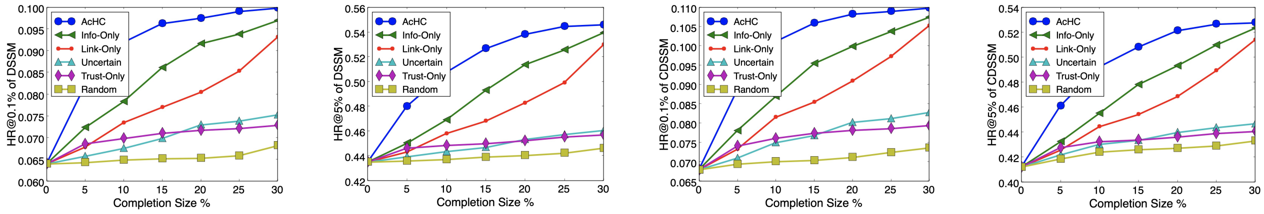
In this section, we present a case study from WeChat, to interpret the Ads-HIN enhanced prediction. More specifically, we divide the ads in different industries and consider the CTR prediction task over these ads. We respectively evaluate the AUC scores of CTR prediction before and after using the HIN features mined from our Ads-HIN, and compute the relative improvement on AUC.

Figure 6 reports the top industries with the highest AUC improvement because of the HIN features. We can see that our method achieves the highest improvement on industries including `Electronics`, `Game`, `Software`, and `Entertainment`. We provide the following interpretations for the results. First, in comparison with other industries (e.g., `Car`), some industries such as `Electronics` and `Entertainment` contain a *diverse* set of ads in many subcategories. For example, the `Electronics` industry may contain ads of high-end products, such as HDTV and low-end ones, such as desk lamp. Thus, the intrinsic features about user profile and ad attributes are too "coarse-grained" to capture user's interest in the ads. In this case, our method can enhance the prediction by incorporating the fine-grained information mined from our Ads-HIN. Second, customers of the industries such as `Game` and `Entertainment` have shown more flexible and dynamic interest. Many game players are found to be always willing to try new games that are released. The intrinsic user features, such as age, gender, game playing history, could hardly describe their changing interest, nor can they predict their behavioral targeting results. In contrast, our method can capture the changing interest of the users by using their behavioral data, such as following official accounts or visiting game forums.

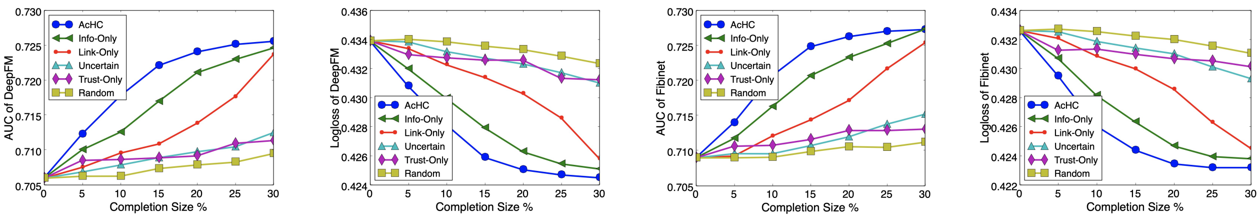
6. RELATED WORK

Behavioral targeting in online advertising. Existing approaches to behavioral targeting can be divided into two categories. The classical methods [28, 3] segment users based on their online behaviors and then deliver the related ads to each user segment. More recently, some behavioral targeting methods [1, 32, 15, 5, 7] are proposed to model the user behavior sequences as the behavior vectors, which is used to directly predict the click behavior. The methods utilized in the work include matrix factorization method [32] that jointly factorizes multiple behavior matrices, standard Bayesian personalized ranking that considers different types of behaviors and neural multi-task learning solutions [7]. Our approach and these works focus on different aspects of behavioral targeting: they focus on specific user behavior modeling strategies, while we pay more attention to incorporate heterogeneous information of user behaviors and ad properties in an HIN to enhance the prediction model.

HIN enhanced recommendation. There are many works to leverage the HIN to improve the recommendation performance. Among them, some methods [25, 10, 31, 24, 4] use the information from the HIN directly to enrich the representation of items or users by applying representation learning techniques to encode the HIN into low-rank embedding. These methods are generally called as embedding-based methods. Other methods [19, 22, 30, 26, 13, 27] lever-

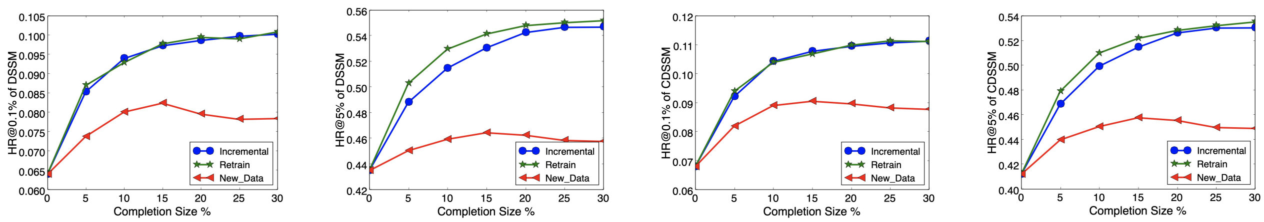


(a) HR@0.1% (DSSM). (b) HR@5% (DSSM). (c) HR@0.1% (CDSSM). (d) HR@5% (CDSSM).

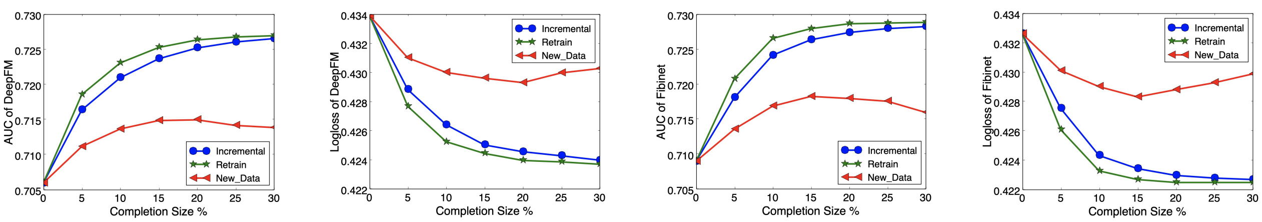


(e) AUC (DeepFM). (f) LogLoss (DeepFM). (g) AUC (Fibinet). (h) LogLoss (Fibinet).

Figure 4: Evaluating Completion Strategies on Ad Matching and CTR Prediction Task.



(a) HR@0.1% (DSSM). (b) HR@5% (DSSM). (c) HR@0.1% (CDSSM). (d) HR@5% (CDSSM).



(e) AUC (DeepFM). (f) LogLoss(DeepFM). (g) AUC(Fibinet). (h) LogLoss(Fibinet).

Figure 5: Evaluating Training Methods on Ad Matching and CTR Prediction Task.

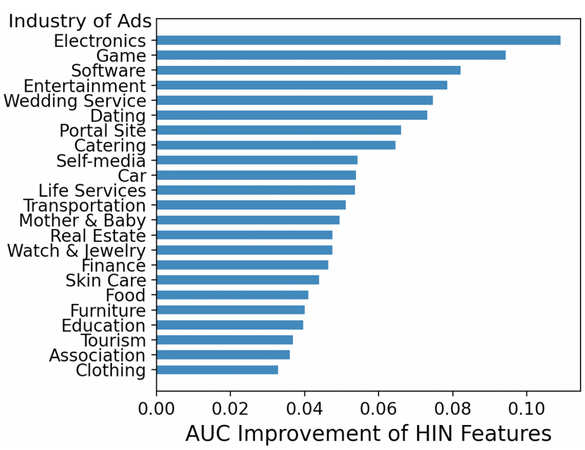


Figure 6: A case study in WeChat advertising.

age the connectivity patterns of the objects in the HIN for recommendation, which are generally called as path-based methods. Among them, [19, 22, 30] integrate matrix factorization with extracted meta-paths in HINs. [26, 13, 27] use deep learning models to encode the path embedding explicitly. Our approach adopts the idea of embedding-based methods studied in the graph representation learning.

7. CONCLUSION

In this paper, we have studied the problem of HIN enhanced behavioral targeting. We introduced an HIN that models different relationships between users and ads from a variety of heterogeneous sources in an advertising platform in WeChat. Then, we extracted "high-quality" features from the HIN and fed the features to a prediction model for behavioral targeting. We devised an active completion algorithm that judiciously leverages human to construct the HIN given budget on number of human involvement. We proposed a utility function for completion tasks answered by human and developed an incremental training method to improve the efficiency of graph representation learning models. We conducted experiments on CTR prediction and ad matching to show performance superiority of our approach.

8. REFERENCES

[1] M. Aly, S. Pandey, V. Josifovski, and K. Punera. Towards a robust modeling of temporal interest change patterns for behavioral targeting. In *Proceedings of the 22nd international conference on World Wide Web*, pages 71–82, 2013.

[2] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[3] T. Chen, J. Yan, G. Xue, and Z. Chen. Transfer learning for behavioral targeting. In *Proceedings of the 19th international conference on World wide web*, pages 1077–1078, 2010.

[4] A. Dadoun, R. Troncy, O. Ratier, and R. Petitti. Location embeddings for next trip recommendation. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 896–903, 2019.

[5] J. Ding, G. Yu, X. He, Y. Quan, Y. Li, T.-S. Chua, D. Jin, and J. Yu. Improving implicit recommender systems with view data. In *IJCAI*, pages 3343–3349, 2018.

[6] S. Fan, Y. Lu, and S. Gupta. Social media in-feed advertising: the impacts of consistency and sociability on ad avoidance. In *PACIS*, page 190, 2017.

[7] C. Gao, X. He, D. Gan, X. Chen, F. Feng, Y. Li, T.-S. Chua, and D. Jin. Neural multi-task recommendation from multi-behavior data. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1554–1557. IEEE, 2019.

[8] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[9] Y. Guo and R. Greiner. Optimistic active-learning using mutual information. In *IJCAI*, volume 7, pages 823–829, 2007.

[10] J. Huang, W. X. Zhao, H. Dou, J.-R. Wen, and E. Y. Chang. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 505–514, 2018.

[11] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338, 2013.

[12] T. Huang, Z. Zhang, and J. Zhang. Fibinet: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 169–177, 2019.

[13] X. Huang, Q. Fang, S. Qian, J. Sang, Y. Li, and C. Xu. Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 548–556, 2019.

[14] S. Jia, Y. Xiang, X. Chen, and K. Wang. Triple trustworthiness measurement for knowledge graph. In *The World Wide Web Conference*, pages 2865–2871, 2019.

[15] B. Loni, R. Pagano, M. Larson, and A. Hanjalic. Bayesian personalized ranking with multi-channel user feedback. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 361–364, 2016.

[16] N. Murray and F. Perronnin. Generalized max pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2473–2480, 2014.

[17] N. Rubens, R. Tomioka, and M. Sugiyama. Output divergence criterion for active learning in collaborative settings. *IPSJ Online Transactions*, 2:240–249, 2009.

[18] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 101–110, 2014.

[19] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2018.

[20] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37, 2016.

[21] C. Shi, Y. Ye, and J. Zhang. Hena 2019: The 3rd workshop of heterogeneous information network analysis and applications. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2991–2992. ACM, 2019.

[22] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 453–462, 2015.

[23] H. Wang, Q. Meng, J. Fan, Y. Li, L. Cui, X. Zhao, C. Peng, G. Chen, and X. Du. Social influence does matter: User action prediction for in-feed advertising. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 246–253, 2020.

[24] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 592–600, 2018.

[25] H. Wang, F. Zhang, X. Xie, and M. Guo. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*, pages 1835–1844, 2018.

[26] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5329–5336, 2019.

[27] Y. Xian, Z. Fu, S. Muthukrishnan, G. De Melo, and Y. Zhang. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the*

42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 285–294, 2019.

[28] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *Proceedings of the 18th international conference on World wide web*, pages 261–270, 2009.

[29] J. Yang, X. Zhao, J. Fan, G. Chen, C. Peng, S. Yao, and X. Du. Behavioral targeting using heterogeneous information networks for social media advertising. In *Technical Report*, 2020. https://github.com/hinsonver/techreport/blob/master/hinbt.pdf.

[30] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 283–292, 2014.

[31] Y. Zhang, Q. Ai, X. Chen, and P. Wang. Learning over knowledge-base embeddings for recommendation. *arXiv preprint arXiv:1803.06540*, 2018.

[32] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi. Improving user topic interest profiles by behavior factorization. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1406–1416, 2015.

[33] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, and D. Cai. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Transactions on Knowledge and Data Engineering*, 32(4):631–644, 2019.

9. APPENDIX

9.1 ADS-HIN Schema

We use the data on the WeChat platform to construct an Ads-HIN. Specifically, we define 8 types of nodes:

- user: the users of WeChat Social Platforms.
- ad: the ads placed in WeChat Moments.
- official_account: the media accounts on WeChat.
- advertiser: the person or company that advertises.
- topic: the topics of ads and official accounts, such as game and beauty.
- brand: the brands related to ads and official accounts, such as Nike and Sony.
- tag: the user's interest tag refined based on the user's basic attributes and behaviors.
- word: the keywords in ad copy, pictures and videos.

Also, we define 11 types of edges, Among the edge types, we consider four types, namely `mention`, `belong_to_topic`, `belong_to_brand` and `advertising_to_tag`, need to be completed by human, as the automatic process cannot provide accurate results for these types. Edges with other types, such as `click` and `follow`, can be automatically constructed based on the data in WeChat.

- click : the relation from a user to an ad, meaning that the user has clicked the ad.
- interact: the relation from a user to an ad, meaning that the user has commented or liked the ad.
- uninterest: the relation from a user to an ad, meaning that the user has blocked or reported the ad.
- advertisting: the relation from an advertiser to an ad, meaning that the advertiser has advertised the ad.
- follow: the relation from a user to an official account, meaning the user has subscribed to the official account.
- interested_in_tag: the relation from a user to a tag, meaning that the user is interested in the tag.
- interested_in_word: the relation from a user to a word, meaning the user is interested in the keyword.
- mention: the relation from an ad to a keyword, meaning that the keyword is mentioned in the text, image and video of the ad.
- belong_to_topic: the relation from an ad or official account to a topic, meaning that the ad or official account belongs to the topic.
- belong_to_brand: the relation from an ad or official account to a brand, meaning that the ad or official account belongs to the brand.
- advertising_to_tag: the relation from an ad to an interest tag, meaning that the ad is targeted advertising to the users interested in the interest tag.

9.2 Proof of Lemma 1

Proof of Lemma 1. Consider the simple graph in Figure 3, let (v_h, v_t, r) be a triplet that is newly added into the graph. The embeddings of each node v in the graph will be updated as the gradient information of (v_h, v_t, r) is propagated through the interlink. We approximate the propagation as the first iteration update of the embedding of v_n caused by (v_h, v_t, r), which is the similar approximate method of the existing active learning work [9].

According to the gradient descent optimizer, TransE updates the embedding of each node v as follows:

$$\mathbf{e}_v \leftarrow \mathbf{e}_v - \gamma \cdot \nabla(\mathbf{e}_v), \tag{9}$$

where γ is the learning rate for back propagation. $\nabla(\mathbf{e}_v)$ is the gradient of $\mathbf{e}_v$, when v is the head or tail node of a triplet, $\nabla(\mathbf{e}_v)$ has the following two different forms:

$$\nabla(\mathbf{e}_v) = \begin{cases} 2(\mathbf{e}_v - \mathbf{e}_o + \mathbf{e}_r) & (v, o, r) \in \mathcal{G} \\ 2(\mathbf{e}_v - \mathbf{e}_o - \mathbf{e}_r) & (o, v, r) \in \mathcal{G}. \end{cases} \tag{10}$$

According to Equation 10, we can rewrite Equation 9 as follows:

$$\mathbf{e}_v \leftarrow \mathbf{e}_v - 2\gamma \cdot (\mathbf{e}_v - \mathbf{e}_o \pm \mathbf{e}_r). \tag{11}$$

Let $\mathbf{e}^+_{v_n}$ denote the updated embedding of node v_n after the completion of triplet (v_h, v_t, r). According to Equation 11, the difference between the updated embeddings of node v_n before and after the triplet (v_h, v_t, r) being completed can be written as:

$$\begin{aligned} \mathbf{e}_{v_n} - \mathbf{e}^+_{v_n} \leftarrow & \mathbf{e}_{v_n} - 2\gamma(\mathbf{e}_{v_n} - \mathbf{e}_{v_{n-1}} \pm \mathbf{e}_r) \\ & - \mathbf{e}_{v_n} + 2\gamma(\mathbf{e}_{v_n} - \mathbf{e}^+_{v_{n-1}} \pm \mathbf{e}_r) \\ = & 2\gamma(\mathbf{e}_{v_{n-1}} - \mathbf{e}^+_{v_{n-1}}) \\ = & (2\gamma)^{n-1}(\mathbf{e}_{v_1} - \mathbf{e}^+_{v_1}). \end{aligned} \tag{12}$$

To obtain the expression of $\mathbf{e}^+_{v_n}$, we need to calculate $\mathbf{e}_{v_1} - \mathbf{e}^+_{v_1}$. We first write the expression of $\mathbf{e}^+_{v_1}$ as follow:

$$\begin{aligned}
\mathbf{e}^+_{v_1} &\leftarrow \mathbf{e}_{v_1} - 2\gamma(\mathbf{e}_{v_1} - \mathbf{e}^+_{v_h} \pm \mathbf{e}_r) \\
&= \mathbf{e}_{v_1} - 2\gamma(\mathbf{e}_{v_1} - \mathbf{e}_{v_h} + 2\gamma(\mathbf{e}_{v_h} + \mathbf{e}_r - \mathbf{e}_{v_t}) \pm \mathbf{e}_r) \\
&= \mathbf{e}_{v_1} - 2\gamma(\mathbf{e}_{v_1} - \mathbf{e}_{v_h} \pm \mathbf{e}_r) - (2\gamma)^2(\mathbf{e}_{v_h} + \mathbf{e}_r - \mathbf{e}_{v_t}) \\
&\leftarrow \mathbf{e}_{v_1} - (2\gamma)^2(\mathbf{e}_{v_h} + \mathbf{e}_r - \mathbf{e}_{v_t}).
\end{aligned} \tag{13}$$

Then we can easily get $\mathbf{e}_{v_1} - \mathbf{e}^+_{v_1} = (2\gamma)^2(\mathbf{e}_{v_h} + \mathbf{e}_r - \mathbf{e}_{v_t})$. Combined with Equation 12, we can obtain the following expression of $\mathbf{e}^+_{v_n}$:

$$\mathbf{e}^+_{v_n} \leftarrow \mathbf{e}_{v_n} - (2\gamma)^{n+1}(\mathbf{e}_{v_h} + \mathbf{e}_r - \mathbf{e}_{v_t}), \tag{14}$$

where n is the length of interlink connecting v_n and v_h. Hence, we prove the lemma.

□