Understanding WeChat User Preferences and "Wow" Diffusion

Fanjin Zhang, Jie Tang*, *IEEE Fellow*, Xueyi Liu, Zhenyu Hou, Yuxiao Dong, Jing Zhang, Xiao Liu, Ruobing Xie, Kai Zhuang, Xu Zhang, Leyu Lin, and Philip S. Yu, *IEEE Fellow*

Abstract—WeChat is the largest social instant messaging platform in China with 1.1 billion monthly active users."Top Stories" is a kind of novel friend-enhanced recommendation engine in WeChat, in which users can read articles based on both their own and their friends' preferences. Specifically, when a user reads an article by opening it, the "click" behavior is private. Besides, if the user clicks the "wow" button, (only) her/his direct connections will be aware of this action/preference. Based on the unique billion-scale WeChat data, we aim to understand user preferences and wow diffusion in Top Stories at different levels. We have some interesting discoveries. For instance, the wow probability of one user is negatively correlated with the number of connected components that are formed by her/his active friends, but the click probability is the opposite. We further study to what extent users' wow and click behavior can be predicted from their social connections. To address it, we present a hierarchical graph representation learning based model ProHENE, which is capable of capturing the structured based social observations discovered above. Our experiments show that the proposed method can significantly improve the prediction performance compared with alternative methods.

Index Terms—Social Influence, Information Diffusion, User Behavior

1 Introduction

INFORMATION diffusion [33] has increasingly changed from offline to online these years. There emerge many popular social applications, such as "News Feed" in Facebook, and "Top Stories" in WeChat, which facilitate information diffusion greatly. Central to information diffusion is the user trait and the social influence between users, which has attracted many researchers working on it [11], [13], [45].

Despite the popular applications and extensive studies of information diffusion algorithms, it is still unclear about the inherent factors that result in different types of user feedbacks in specific social contexts. First, how can user attributes, the relations between users and friends, and the local network structure influence user behavior? Second, what are the differences between various kinds of user feedbacks (such as "click", "like" and "share") w.r.t. above factors? Such problems are still largely unexplored and far from understood.

In this work, we study the "Top Stories" service in WeChat—the largest social instant messaging platforms in China—to understand the user behavior and the connections between user behavior and social relationships. In Top Stories, a user can see the articles wowed by her friends, which can be regarded as **share** plus **like**, and she can perform the wow or click action on each article as well. An illustration example of WeChat's Top Stories service is shown in Figure 1, in which each user will be shown the articles that her friends wowed as well as those friends' names, and she can click into the article or also wow it. Herein, we aim at understanding users' wow and click behavior from different aspects, including user demographics, social relationships, and users' ego network structures.

The research problem in this paper is related to social influence locality [49], which targets at quantifying how

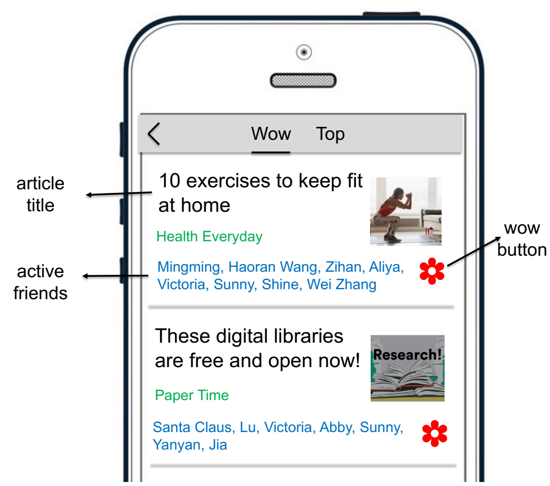


Fig. 1. "Top Stories" in WeChat. Each user can view what friends "wow"ed. If she/he also wows one article, it will be displayed to (only) her/his friends, which forms a diffusion process. Here "active friends" means friends who wowed the corresponding articles.

- *Fanjin Zhang, Xueyi Liu, Zhenyu Hou and Xiao Liu are with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China. E-mail: {zfj17, xueyi-li18, hzy17, liuxiao17}@mails.tsinghua.edu.cn*
- *Jie Tang is with the Department of Computer Science and Technology, Tsinghua University, and Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, 100084, China. E-mail: jietang@tsinghua.edu.cn, corresponding author*
- *Yuxiao Dong is with Microsoft Research, Redmond, WA 98052, USA Email: ericdongyx@gmail.com*
- *Jing Zhang is with the Renmin University of China, Beijing, 100872, China. Email: zhang-jing@ruc.edu.cn*
- *Ruobing Xie, Kai Zhuang, Xu Zhang and Leyu Lin are with the WeChat Search Application Department, Tencent, Beijing, 100080, China. Email: {ruobingxie, terryzhuang, xuonezhang, goshawklin}@tencent.com*
- *Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA. Email: psyu@uic.edu*

user behavior is influenced by other users in the ego networks, and more broadly related to social influence [38] and information diffusion [26], [34]. Most methods [23], [49] address the social influence locality problem by using handcrafted user features and network features to predict user behaviors. Recently, Qiu et al. [31] propose to use graph attention networks (GAT) to learn user proximity in ego networks. Wang et al. [43] further consider topological features through Weisfeiler-Lehman (WL) algorithm [7] to predict user behavior for in-feed advertising. However, they haven't dug into studying potential influence factors in different granularities, such as user demographics, and ego network properties, based on which better prediction models can be designed.

Through the study, we first reveal several intriguing discoveries that impact user behavior at different levels. Based on the discoveries, we then develop a unified framework to predict users' wow and click behavior by modeling user attributes, user relations and ego network structures.

To highlight several of our key findings, for user demographics, we find that users' wow and click behavior varies by gender and age, and the patterns become complicated when cross-attribute factors are considered. For user relations, users are likely to behave differently when their active friends are structural holes and opinion leaders. Considering ego network properties, both wow probability and click probability are strongly correlated with the number of connected components formed by users' active friends, but they have the opposite patterns. This correlation becomes stronger when the ego network is cleaned.

Based on these interesting discoveries, we further study to what extent users' behavior can be predicted from their social connections and attributes. To this end, we propose a hierarchical graph representation learning based model ProHENE. Our model is closely related to and motivated by the insights from data, which is different from many neural network based methods. Specifically, first, **to model cross-attribute factor for users' different attributes as the analysis in Section 3.1** (such as user embedding and demographics), we adopt factorization machine technique to generate second-order features to model feature interactions for each individual. Second, **to remove noises in the ego networks as the analysis in Section 3.3**, ProHENE propagates initial user features in the modulated spectral domain, to generate user embeddings based on cleaned ego networks. Third, **to model user relationships as the analysis in Section 3.2**, we adopt a new graph attention mechanism to model feature interactions between neighbors. Fourth, **to model the connected components — hierarchical structure of the ego networks as the analysis in Section 3.3**, we generate hierarchical representations of ego networks by clustering nodes together and learning on the coarsened graphs iteratively. We evaluate the proposed method on large-scale WeChat Top Stories dataset and a public Weibo dataset. Our experiments show that the proposed solution can consistently outperform alternative methods.

2 Background — Top Stories Dataset

Different from other news feed systems, Top Stories in WeChat will recommend to a user those articles favored (wowed) by her/his friends. As the example shown in Figure 1, the recommended articles to the current user are two articles favored by her active friends. The user can choose to click the "wow" button so that her friends will also be informed with the favorite. In this way, the "wow" essentially plays an implicit diffusion role. On the other hand, the user can also click to view the full content of the article or simply ignore it. The dataset used in this paper is the complete dataset of Top Stories generated from Oct 1 to Dec 31, 2019. It consists of three parts: 1) a social network $G = \{U, E\}$, where U is a set of users, and E represents a set of edges recording friendships between users; 2) user attributes C including users' gender, age, regions and so on; and 3) the interaction between users and articles $L = \{(u, d, ts, is\_like, is\_click, af(u, d, ts))|u \in U, d \in \mathcal{D}\}$, where u is the ego user, d is a displayed article in article set $\mathcal{D}$, ts is the timestamp, is_like and is_click are whether u wows and clicks d, respectively, $af(u, d, ts)$ is u's active friends who wowed d before timestamp ts. Note that an interaction can be represented by a triplet (u, d, ts). To avoid over-fitting, we select a subset of the data by first extracting users who performed at least ten interactions (wow or click) and then extracting these users' friendship network and their attributes. The final dataset contains 48,084,772 users, 61,252,317 articles, and 7,459,660,092 interactions.

To start with our analysis, we first provide several necessary definitions which will be used later.

Definition 2.1. Connected Components (CC)[1]. In graph theory, a connected component of an undirected graph G is an induced subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the rest of the graph. We let **CC** denote the acronym of *connected components* and **#CC** denote the number of connected components in a graph.

Definition 2.2. Structural Hole (SH) [2]. A "structural hole" is a term for recognizing a missing bridge in a graph. Wherever two or more groups fail to connect, one can argue that there is a structural hole, a missing gap waiting to be filled. We let **SH** denote the acronym of *structural hole*.

Definition 2.3. Ego network and τ-ego network. Ego network $G_u = \{U_u, E_u\}$ is a subgraph of a static social network G centered at the focal node ("*ego*"), where U_u is the node set consisting of the ego and its first-order neighbors, E_u is the edges between nodes in U_u in the original graph G. **τ-ego network** $G_u^\tau = \{V_u^\tau, E_u^\tau\}$ is a subgraph induced by u and u's τ-degree friends, V_u^τ is the node set of the subgraph G_u^τ and E_u^τ is the edge set of G_u^τ.

Definition 2.4. Active Friends. In this article, we define active friends as the friends who performed "wow" action on an article. In WeChat Top Stories, if a user wowed an article, his/her friends will be informed about it. However, users' click behavior will not shown to the friends directly. As shown in Figure 1, the names shown below the articles are the friends who performed "wow" action.

Definition 2.5. Active Rate. We define **"active rate"** to refer to both wow probability and click probability.

1. https://en.wikipedia.org/wiki/Component_(graph_theory)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3064233, IEEE Transactions on Knowledge and Data Engineering

TABLE 1
User activity w.r.t. gender.

Gender	Wow prob.	Click prob.
Male	1.17%	**10.62%**
Female	**1.19%**	9.86%

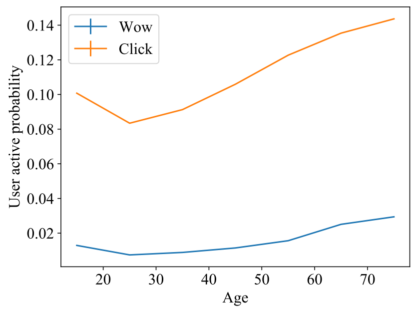
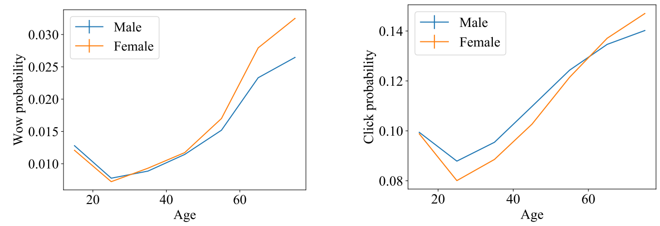


Fig. 2. Wow and click probability w.r.t. user age.



(a) Wow Behavior (b) Click Behavior

Fig. 3. User wow and click probability w.r.t. users' gender and age.

TABLE 2
Pairwise influence w.r.t. the user and the friend's gender.

User	Friend	Wow prob.	Click prob.
Male	Male	0.97%	**11.19%**
Male	Female	**1.01%**	9.69%
Female	Male	0.93%	9.11%
Female	Female	**1.06%**	**10.33%**

3 Analysis and Discoveries

Based on the large-scale interactions between users and articles, we investigate how users' wow and click behavior correlates with three aspects: (1) user demographics, (2) user relationships, and (3) ego network properties. Next, we present these analysis results one by one.

3.1 User Demographics

Table 1, Figure 2, and Figure 3 show the probability that people of different gender and age wow or click articles in Top Stories. From Table 1, we can observe that males' click probability is clearly higher than females', while females' wow probability is a little bit higher than males'. The reason might be that males tend to consume content, but females are more active in social circles. Regarding age, the patterns are very interesting. According to our intuition, young generations (users of the 20s and 30s) are the most active users in our online social circles. However, in Figure 2, the wow and click probability of the 20s and 30s is the lowest among all ages. We infer that young people might too busy to look at the articles in detail or their "reverse psychology" reacts to the recommended articles. Moreover, when we consider both gender and age attributes, the patterns become different again. In Figure 3, we find that for people younger than 20s, males are more active than females. However, there is a reversion for both wow and click behavior, but at different split points (40s for wow and 60s for click), indicating that older female users are more active than older male users. The result demonstrates that the cross-attribute factor is more complicated.

3.2 User Relationships

For user relationships, we consider pairwise relations and triangle relations between users and their active friends. To eliminate other influence factors, we consider interactions with only one friend who wowed the article for pairwise relations, and exactly two active friends for triangle relations. We analyze user relationships from two views: user demographics and user social roles.

Pairwise Relations w.r.t. User Demographics. Table 2 shows the users' active rate concerning pairwise relations between ego users' gender and friends' gender. We observe that for click behavior, when friends' gender is the same as ego users', the ego users' click probability is higher, which can be explained by the homophily of user interests. However, for wow behavior, users are more likely to wow an article when their active friends are females.

In view of ages, Figure 4 visualizes users' wow probability w.r.t. pairwise relations between users' ages and friends' ages. We have several interesting discoveries. First, when users are young (< 40 years old), they are more influenced by their older friends than friends of the same age group. Second, older users are highly influenced by their friends of the same age group. Meanwhile, they also care about what young generations wowed, which shows the cross-generation care, such as parents to children, managers to subordinates, etc. The pattern of click behavior is omitted here since it is similar to that of wow behavior.

Talking about the region, we also consider the distance between users and their friends (incorporating users' regions). Table 3 shows the users' active rate w.r.t. the distance between the user and the active friend. We can see that when the geographic distance between the ego user and the friend is close, the wow probability and click probability of the ego user is higher, which shows the existence of interest homophily w.r.t. user region.

Pairwise Relations w.r.t. User Social Roles. We also study pairwise influence between users' social roles and friends' social roles. Here social roles refer to users' roles in the social network or wow diffusion network. Table 4 shows the

TABLE 3
Pairwise influence w.r.t. the distance between the user and the friend.

User	Wow prob.	Click prob.
All	1.01%	10.24%
Same province	1.05%	10.65%
Same city	1.08%	10.85%
Same district	**1.19%**	**11.27%**

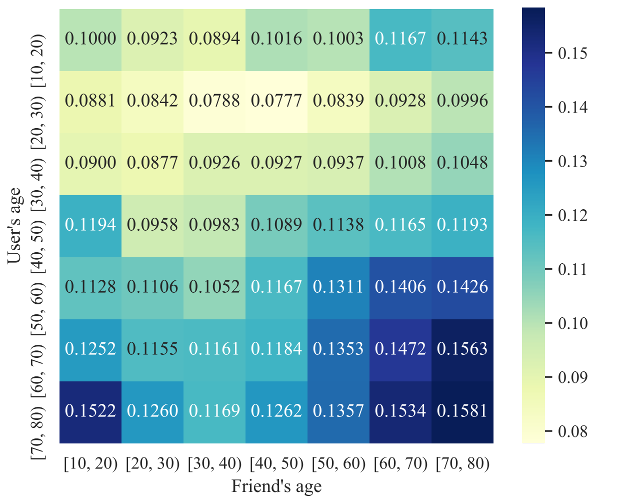


Fig. 4. Users' wow probability w.r.t. the user's age and the friend's age.

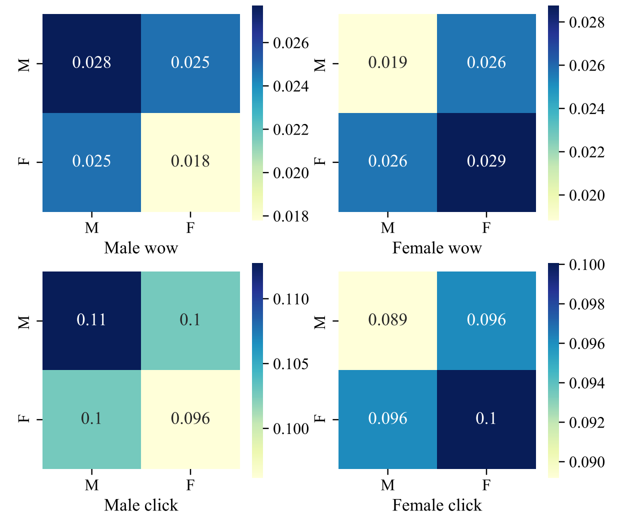
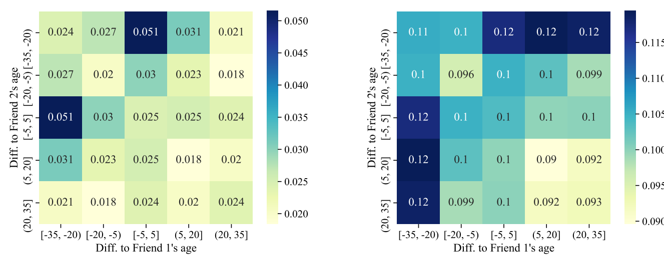


Fig. 5. Users' wow and click probability w.r.t. the user's gender and friends' gender. M: Male, F: Female.

TABLE 4
Pairwise influence w.r.t. users' and friends' social roles. OU: ordinary user; OL: opinion leader.

User	Friend	Wow prob.	Click prob.
OU	OU	**1.06%**	**10.65%**
OU	OL	0.66%	8.11%
OL	OU	**0.92%**	**8.57%**
OL	OL	0.64%	7.49%



(a) Wow Behavior (b) Click Behavior

Fig. 6. User wow and click probability w.r.t. the differences between users' age and friends' age. Here the difference is calculated by "the age of the friend minus the age of the ego user".

users' active rate w.r.t pairwise relations of different social roles: opinion leaders (OL) and ordinary users (OU). We find opinion leaders in social networks by running PageRank algorithm [28] on user diffusion network and then regard users with top 1% PageRank scores as opinion leaders. Surprisingly, we find users' wow and click probability is higher when their active friends are not opinion leaders. For wow behavior, the reasons might be that if an opinion leader has wowed one article, the ego user is less willing to publicize it because many users connected to opinion leaders probably already knew it. For click behavior, we think perhaps users browse Top Stories mainly for recreation as WeChat is a social instant messaging platform for friends and acquaintances. Therefore, users might care more about what similar friends are interested in rather than what opinion leaders pay attention to.

Besides, Table 5 shows the users' active rate w.r.t. pairwise relations of different social roles: structural holes (SH) and ordinary users (OU). Here we find cut points in users' friendship network using Tarjan [40] algorithm to approximate structural hole users. Clearly, users' wow behavior is highly influenced when their friends are structural holes, which demonstrates that structural holes are critical in the information diffusion process. Also, the wow and click probability of ordinary users is higher if their active friends are structural holes. For users who are structural holes, their click probability is higher when their friends are not structural holes, but the difference is not very significant.

TABLE 5
Pairwise influence w.r.t. users' and friends' social roles. OU: ordinary user; SH: structural hole.

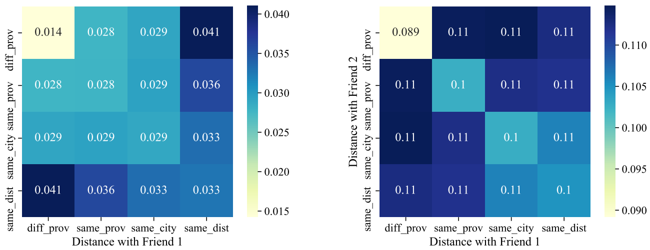
User	Friend	Wow prob.	Click prob.
OU	OU	0.99%	10.26%
OU	SH	**1.38%**	**12.43%**
SH	OU	2.34%	**10.28%**
SH	SH	**3.58%**	10.16%

Triangle Relations w.r.t. User Demographics. Here we study how triangle relations — more complex relationships, would influence ego users' behavior. For triangle relations, we consider interactions with exactly two friends who wowed the article. We analyze the demographics (i.e., gender, age, and region) of the ego users and their friends.

Figure 5 shows the users' active rate with respect to triangle relation between ego user's gender and his/her two friends' gender. From the figure, we can observe consistent patterns of wow and click behaviors. If the two friends' gender is the same as the ego user's gender, the ego users' active rate is highest. Again, this implied a high degree of gender homophily.

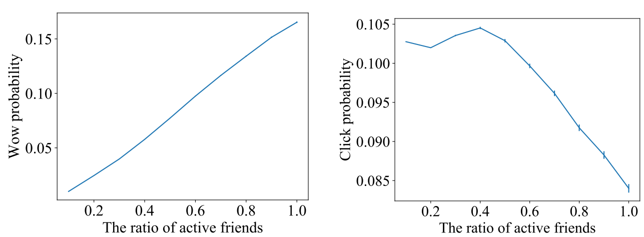
Furthermore, Figure 6 shows the users' active rate w.r.t. the difference between the ego user's age and the two friends' ages. We discover that if one friend is of the same age group and the other friend is younger than ego user, the active rate of ego user is high. Furthermore, the color in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3064233, IEEE Transactions on Knowledge and Data Engineering



(a) Wow Behavior (b) Click Behavior

Fig. 7. User wow and click probability w.r.t. the distance between the ego user and the friends. Here "diff_prov" means different provinces, "same_prov" means the same provinces, "same_city" means the same cities, and "same_dist" means the same districts.



(a) Wow Behavior (b) Click Behavior

Fig. 8. User wow and click probability v.s. The ratio of active friends

the left top is darker than that of the right bottom, which demonstrates that old users pay more attention to young users compared with young users attending to old users.

Additionally, Figure 7 shows the users' active rate w.r.t. the distance between the ego user and the two active friends. Our intuition might be that if the distance between the ego user and her two friends is close, the ego users' active rate is high. However, actually, if one friend is nearby, and the other friend is more distant from the ego user, the ego user will be more active. This kind of "attribute diversity" may provide evidence that the wowed articles are acknowledged by various users.

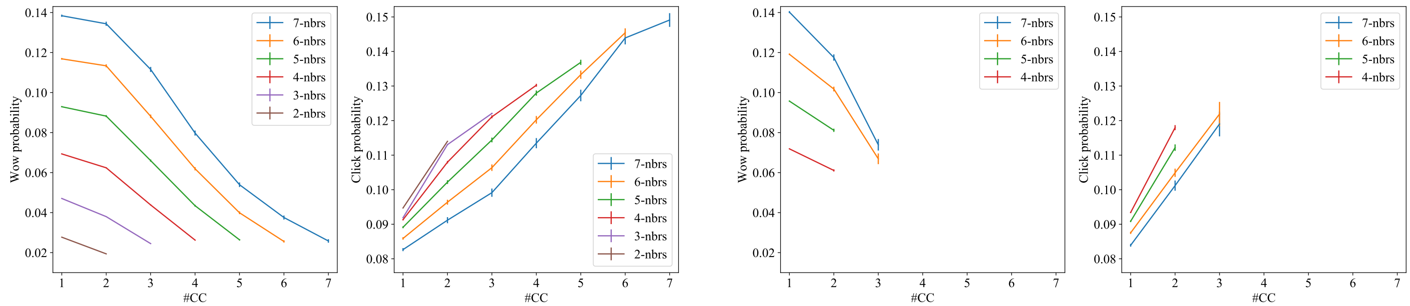
3.3 Ego Network Property

In this subsection, we study the correlation between users' activity and their ego network property. To be precise, the ego network is defined as the induced subgraph of users' active friends. We find that users' online behaviors (click and wow) are strongly influenced by their friend circles (users in their ego networks). We study ego network properties from three aspects: the number of friends in the ego network, the number of connected components (#CC) in the ego network, #CC in the cleaned ego network (k-core subgraph).

The Number of Friends in the Ego Networks. Figure 8 shows how a user's wow and click probability on an article changes when the number of active friends increases. We define the ratio of active friends by dividing a pre-defined maximum number of friends into the actual number of friends. It demonstrates two very different patterns w.r.t. the two behaviors. For wow behavior, with the number of active friends increasing, the probability that the user wows an article also increases, roughly linearly, while for click behavior, the probability first fluctuates a little, and then decreases clearly after the ratio of one's active friends increasing to 0.4. The phenomenon could be explained by information overload — when the number of one' active friends is large, the user may have many other channels from these friends to learn about the information, such as "Moments" or "Subscriptions" (the first is reading articles posted by friends and the other is reading articles of subscribed accounts), thus losing interest in clicking it [9]. For wow behavior, the pattern is consistent with our intuition that people may continue to share the hot spots to let more and more people care about them.

The Number of Connected Components (#CC) in the Ego Network. Following the above, we conduct another deep analysis, named structural diversity [41], to study how the topological structure of one's active friends would influence the user's behavior. Figure 9(a) plots the probability of a user's behavior w.r.t. the number of connected components (#CC) of her/his active friends. Here, each connected component can be viewed as a specific group of friends (connected by friendships among them). The pattern is very interesting. When the total number of active friends on an article increasing, the user may increase the probability of spreading the article (see Figure 8). However, when fixing the number of active friends, ranging from 2 to 7, the probability decreases with the increase of the number of connected components (#CC) (see Figure 9(a)). This confirms the structural diversity analysis in sociology [41], [49], which suggests that the user's interest in sharing a piece of information will decrease when she/he notices that the information has already been shared by multiple different groups of friends, since there isn't much benefit for growing the user's influence when many people have shared it. For click, it is totally different — when a user notices that she/he has multiple different groups of friends read an article, her/his probability of reading the articles will quickly increase. If many friends of different circles have wowed one article, the article quality is probably high and has a broad audience, so the user is attracted to read it.

#CC in the Cleaned Ego Network. Although one can have many active friends who wowed an article, different friends may influence the ego user to different extent. For example, friend O and the ego user perhaps became friends at a chance, and they are not familiar with each other. Thus, O is an outlier in the ego network who doesn't connect to other friends of the ego user. At this time, including friend O in the ego network may introduce noises. Thus, we want to first obtain the cleaned ego network, and then analyze the correlation of its structure and ego user's activity. To obtain the cleaned ego network, we extract the 1-core subgraph of the ego network formed by active friends, where 1-core means the node in the subgraph needs to have at least one edge with other nodes. In this way, outliers are removed from the ego networks. Figure 9(b) plots the probability of a user's behavior w.r.t. #CC of one's 1-core subgraph of the ego network formed by active friends. Note that 1-core ensures the connectivity of nodes in the subgraph, so some combinations of (#Friends, #CC) pairs don't exist, such as 7 friends with 7 CC. Comparing Figure 9(b) with Figure 9(a), we can see that when fixing the number of active friends (such as 7), the speed of increase or decrease of wow/click



(a) Users' Active Rate v.s. #CC in the Ego network formed by active friends

(b) Users' Active Rate v.s. #CC in the 1-core subgraph of the ego network formed by active friends

Fig. 9. Social influence analysis: the probability that the user wows or clicks an article conditioned on the number of connected components of (cleaned) ego networks formed by active friends.

probability w.r.t. #CC in Figure 9(b) is obviously faster. This difference shows that the structural topology of cleaned ego networks probably gives a better discriminative ability to predict ego users' activity.

3.4 Summaries

From the above analysis, we have the following discoveries:

- *Males are more likely to click but less likely to wow the articles than females. Counterintuitively, the young generations (people of 20s and 30s) have the lowest active rate in Top Stories.*
- *For pairwise user relations, there exists interest homophily between users and friends (such as about gender and region), but attribute diversity (such as region) also positively correlates with users' activity when there is more than one active friend.*
- *According to ego network topology, the patterns of wow and click behavior are very different. For instance, when fixing the number of active friends, users' wow probability is negatively correlated to #CC formed by active friends, but for click behavior, it is the opposite. The patterns can be more significant when the ego network is cleaned.*

4 Predictive Model

Can we leverage the discovered patterns to predict users' online behaviors? In this section, we first briefly formulate the problem and then present our prediction framework.

4.1 Problem Formulation

Let $G_u^\tau = \{V_u^\tau, E_u^\tau\}$ be user u' s τ-ego network where **τ-ego network** is a subgraph induced by u and u's τ-degree friends, V_u^τ is the node set of the subgraph G_u^τ and E_u^τ is the edge set of G_u^τ. The attribute matrix of users in V_u^τ is denoted as C_u^τ. When user u is displayed with an article d wowed (shared) by some friends at timestamp ts, we denote an action status of user u's ego-network as $S_{(u,d,ts)} = \{s_{(v,d,ts)} \in \{0,1\} | v \in V_u^\tau \setminus \{u\}\}$, where $s_{(v,d,ts)}$ is the action status of user v w.r.t. article d before timestamp ts, here 0 means inactive and 1 means active (both denoting wow behavior). Our goal is to quantify the wow and click probability of ego user u after timestamp ts:

$$P(s_{(u,d,>ts)} | G_u^\tau, S_{(u,d,ts)}, C_u^\tau) \tag{1}$$

Since we analyze two different behaviors (click and wow) of users, a straightforward idea is to leverage the correlation between click and wow to design a joint prediction model (like multi-task learning). However, there is no evident correlation between click and wow in our training set. According to our statistics, $P(is\_click_u = 1) \approx P(is\_click_u = 1 | is\_wow_u = 1)$, which means that the two behaviors are almost independent. Thus, we choose to learn independent models for predicting the two behaviors. In the following, we will illustrate our model framework in detail.

4.2 The ProHENE Framework

In this subsection, we present our proposed model framework ProHENE as illustrated in Figure 10. The core model components and the basic idea are as follows: (1) For input user features, we consider various user features such as user demographics (gender and age) and pre-trained user embeddings, and try to model feature interactions as the analysis in Section 3.1. (2) We then learn user embeddings by propagating initial features in a trainable modulated spectral domain, by which the learned user embeddings can capture useful information in ego networks and filter out those noises, which is motivated by the analysis in Section 3.3. (3) Next we further feed the learned intermediate representations to a hierarchical graph representation model. This model can learn subgraph embeddings by clustering nodes iteratively (here subgraphs can be considered to correspond to connected components analyzed in Sec. 3.3). (4) Besides, we try to model the interactions between users' features and friends' features with a new attention model as the analysis in Section 3.2. The proposed ProHENE framework consists of five steps: Preprocessing ego networks, Input layer, Feature smoothing layer, Hierarchical graph representation learning and Output layer.

Preprocessing Ego Networks. As the τ-ego network can be very large in such a dense social network, especially for users with large degrees, we adopt a sampling strategy to sample a subset of users from one's ego-network. In this work, we use Breadth-First Search (BFS) to generate fixed-size ego network for each user/interaction due to its effectiveness and efficiency. In detail, we first add the ego user and active friends into the ego network in order. The added order of active friends is determined by their active

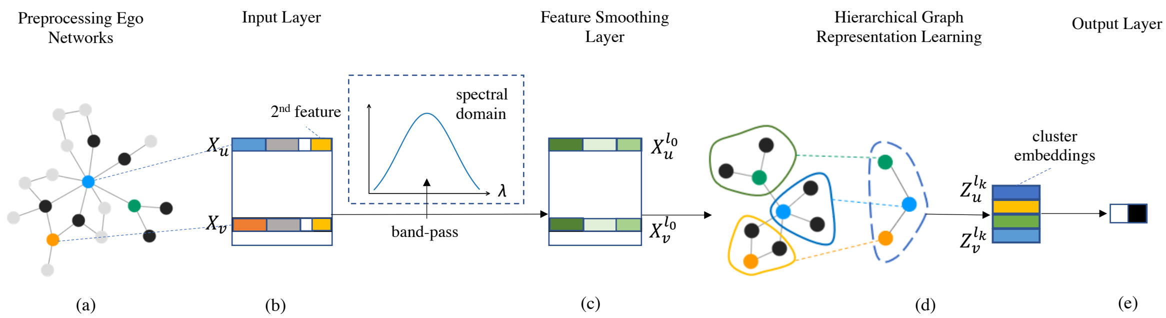


Fig. 10. Model Framework: (a) A sketch map of the processed ego network; (b) The input layer, in which each user's pretrained embedding is concatenated with handcrafted features and her influence feature which indicates her active status and whether she is the ego user; (c) The feature smoothing layer, in which the each user's concatenated input features are filtered by a band-pass filter in the spectral domain; (d) Each ego network's features are passed into a hierarchical graph representation learning model; (e) The output layer.

TABLE 6
List of input features of ProHENE. (*x) indicates the dimension of the feature. OL: opinion leader. SH: structural hole.

Type	Description	Feature Definition
Demographics	Gender	(*1) 0: unknown, 1: male, 2: female
	Age	(*1) Age number
	Region	(*10) Encoding via region partition
Social Roles	OL	(*1) PageRank score
	SH	(*1) Cut point or not
Context	Ego	(*1) Ego user or not
	Action	(*1) wow or not
Embedding	Pre-train	ProNE embedding
Cross Feature	2nd feature	FM w.r.t. various features

(wow) timestamp. Then, the friends of users in the current ego networks are added by performing BFS iteratively. Note that Qiu et al. [31] use Random Walk with Restart (RWR) [14] to generate the sampled ego networks. However, information diffusion in WeChat is very localized (users can only see articles their friends wowed); thus, BFS is more suitable than RWR in this case. Finally, we perform BFS in users' 2-ego networks ($\tau = 2$). We set the number of nodes in each sampled ego network as m. The adjacency matrix of the generated ego network for each instance is denoted as $A_{(u,d,ts)}$ (here an instance refers to an interaction between user u and article d before timestamp ts). We omit the subscripts (u, d, ts) in the following description if there is no ambiguity.

Input Layer. We consider various types of input features for each user, as listed in Table 6. First, the input layer covers customized user features, such as demographic and social role features. Second, we also consider two-dimensional contextual features for each user to indicate the active status and positions in the ego network, in which one is whether the user wowed the corresponding article, and the other is whether the user is the ego user [31]. Third, we further consider pre-trained user embeddings. Although we study user behavior prediction in the ego network, it would be beneficial to capture user structure information in the global social network. Thus, we first pre-train user embeddings in the large-scale friendship network. Many network representation learning methods [29], [37], [48] have been proposed to learn node representations in a graph. We adopt ProNE [48] to pre-train user embeddings in the large-scale friendship network due to its high efficiency and effectiveness, which could take a relatively short time to generate node embeddings of billion-scale graphs and is effective by capturing global information by propagating embeddings in the spectrally modulated domain.

The above mentioned features can be regarded as first-order user features. Motivated by the analysis in Subsection 3.1, the cross-attribute factor might also take effect. Thus, we adopt factorization machine technique to model feature interactions. We generate second-order features by first mapping different features into the same space, and then calculate the second-order feature interactions as follows:

$$X^{\text{2nd}} = \frac{1}{2}((\sum_{i=1}^{F} W_i x^{(i)})^2 - \sum_{i=1}^{F}(W_i x^{(i)})^2) \quad (2)$$

where $x^{(i)}$ is i^{th} user feature, W_i is the feature projection matrix of feature $x^{(i)}$ and F is the number of different features. Here the cross terms indicate the interactions between different features. Finally, we concatenate all the first-order features $\{x^{(i)}\}_{i=1}^{F}$ and the second-order feature X^{2nd} to form the input feature X.

Feature Smoothing Layer. Pre-trained user embeddings only capture the global network structure. Users residing in different ego networks may play different roles. Thus, they should have different representations in different ego networks. We propose a feature smoothing method via graph filters, which can fine-tune user embeddings X via cleaned ego network structures. The output of this step is X^{l_0} by propagating X in the modulated spectral domain of ego networks.

In graph theory, the random walk normalized graph Laplacian is defined as $\mathcal{L} = I_m - D^{-1}A$, where A is the adjacency matrix of the ego network, m is the size of each ego network, I_m is the identity matrix, and $D = \sum_j A_{ij}$. The normalized Laplacian can be decomposed as $\mathcal{L} = U\Lambda U^{\top}$, where $\Lambda = \text{diag}[\lambda_1, \lambda_2, ...\lambda_m]$. In spectral graph theory, small (large) eigenvalues in graph Laplacian control the network's global clustering (local smoothing) effect, which motivates us to capture useful information of ego network in the spectral domain. Global clustering (local smoothing) effect means how well a graph can be partitioned into a small (large) number of clusters so that nodes in different clusters are less connected while nodes in the same clusters are densely connected. The smaller the jth eigenvalue λ_j is, the better partition effect it would achieve for dividing to j clusters [48]. Thus we employ a graph filter g to adjust eigenvalues of the Laplacian.

$$\widetilde{\mathcal{L}} = U\text{diag}([g(\lambda_1), g(\lambda_2), ..., g(\lambda_m)])U^\top \tag{3}$$

where $\widetilde{\mathcal{L}}$ is the modulated Laplacian and g is the spectral modulator. We propagate the initial node embeddings in the spectral domain via modulated Laplacian as follows.

$$X^{l_0} = D^{-1}A(I_m - \widetilde{\mathcal{L}})X \tag{4}$$

where X^{l_0} is the user embedding matrix after modulated in the spectral domain. Here $I_m - \widetilde{\mathcal{L}}$ is the spectral modulator of the normalized adjacency matrix $D^{-1}A$. In this paper, we adopt the graph filter as follows:

$$g(\lambda) = e^{-\frac{1}{2}[(\lambda-\mu)^2-1]\theta} \tag{5}$$

where g can be considered as an adjustable band-pass filter kernel (see Figure 10) with $\mu \in [0, 2]$. It passes or enlarges eigenvalues within a certain range and filters out the other values, thus reducing the noise or redundant information.

To avoid explicit eigendecomposition and Fourier transform, we use the same trick in [48] to approximate g with Chebyshev expansion and Bessel function [1]. In our model, we set μ as a trainable parameter. Thus it can be adaptively learned for different datasets.

Hierarchical Graph Representation Learning. As we analyze above, the ego user's activity is strongly correlated to the number of connected components of her ego-network formed by active friends (neighbors). Our idea here is to design a hierarchical representation learning to encode the substructures of ego networks. Substructures, such as connected components, can be regarded as high-level structural patterns, which motivates us to cluster similar nodes iteratively to encode these substructures. The goal of this step is to generate high-level ego network representation Z^{l_k} at iteration k. We employ graph neural networks (GNN) [27], [36], [46] as basic modules to learn graph representation.

In detail, we first generate node embeddings of the entire ego networks via a GNN [18], [42] module. The input user embeddings are from the learned user embeddings via graph filters described above.

$$Z^{l_1} = \text{GNN}_{0,\text{embed}}(A^{l_0}, X^{l_0}) \tag{6}$$

where Z^{l_k} is the hidden node embedding of layer l_k, A^{l_0} is the adjacency matrix of ego networks, and X^{l_0} is the input node features. In order to generate coarsened graphs to represent graph substructures, following DIFFPOOL [46], we learn an assignment matrix $B^{l_{k+1}}$ via another GNN,

$$B^{l_{k+1}} = \text{softmax}(\text{GNN}_{k,\text{pool}}(A^{l_k}, X^{l_k})) \tag{7}$$

where $B^{l_{k+1}} \in R^{m_k \times m_{k+1}} (m_{k+1} < m_k, m_0 = m)$ and $b_{ij}^{l_{k+1}}$ represents the probability of assigning node i to j'th clusters in $(k+1)$'th assignment layer .

With assignment matrix B^{l_k}, ego-network can be transformed to a smaller graph iteratively, in which each node represents a "cluster".

$$X^{l_k} = {B^{l_k}}^\top Z^{l_k} \in \mathbb{R}^{m_k \times h_k} \tag{8}$$

$$A^{l_k} = {B^{l_k}}^\top A^{l_{k-1}} B^{l_k} \in \mathbb{R}^{m_k \times m_k} \tag{9}$$

where X^{l_k} is the cluster embeddings, and A^{l_k} is the coarsened adjacency matrix which denotes the connectivity strength between pairwise clusters.

Based on the coarsened graph, the coarse-level of node embeddings can be generated by

$$Z^{l_{k+1}} = \text{GNN}_{k,\text{embed}}(A^{l_k}, X^{l_k}) \tag{10}$$

We generate different levels of (sub)graph embeddings by pooling operations on node embedding matrices, and then concatenate them to form the final representations of ego networks,

$$Z^{\text{graph}} = \Big\Vert_{k=1}^{L} \sigma(Z^{l_k}) \tag{11}$$

We set σ in Eq. 11 as a dimension-wise max-pooling operation or sum-pooling to transform node embedding matrix to graph embedding.

Basic GNN modules. As for GNN modules in Eq. 6, Eq. 7 and Eq. 11, we argue that GAT [42] is suitable since it can learn the attention weights of neighboring nodes with their node features. GAT learns the attention weight between node i and node j as follows:

$$\alpha_{ij}^{\text{AA}} = \frac{\exp(\text{act}(a_{\text{src}}^\top W_p x_i + a_{\text{dst}}^\top W_p x_j))}{\sum_{t \in \mathcal{N}_i} \exp(\text{act}(a_{\text{src}}^\top W_p x_i + a_{\text{dst}}^\top W_p x_t))} \tag{12}$$

where $\mathcal{N}_i$ is the neighbors of node i, x_i is the hidden embedding of node i, W_p is the feature projection matrix, a_{src} and a_{dst} are attention parameter, and act is the LeakyReLU activation function. As shown in Eq. 12, GAT uses *additive attention (AA)*.

However, the attention mechanism used in GAT doesn't consider feature interactions between neighboring nodes. Thus, we employ a simple modification over Eq. 12:

$$\alpha_{ij}^{\text{DA}} = \frac{\exp(\text{act}((a_{\text{src}}^\top W_p x_i + b_{\text{src}}) \cdot (a_{\text{dst}}^\top W_p x_j + b_{\text{dst}})))}{\sum_{t \in \mathcal{N}_i} \exp(\text{act}((a_{\text{src}}^\top W_p x_i + b_{\text{src}}) \cdot (a_{\text{dst}}^\top W_p x_t + b_{\text{dst}})))} \tag{13}$$

where b_{src} and b_{dst} are bias terms. We term this attention as *Dot attention (DA)*. As shown in Eq. 13, the cross terms could model feature interactions of neighboring nodes. We denote our model using GAT modules as $\text{ProHENE}_{\text{AA}}$ and the model using dot attention as $\text{ProHENE}_{\text{DA}}$.

Output Layer. Finally, we let the ego network embedding Z^{graph} pass into fully connected layers to generate the prediction scores, which is used to compare it with the ground-truth wow/click labels. We use the cross-entropy loss as our objective function. The prediction function at the output layer and the loss function are described in Eq. 14 and Eq. 15, respectively, where fc_{pred} reprensents the fully-connected layers, y_{pred}^i denotes the action probability of instance i and y_i is the ground-truth of instance i.

$$y_{\text{pred}} = fc_{\text{pred}}(Z^{graph}) \tag{14}$$

$$\mathcal{L} = -\sum_{i=1}^{N} (y_i \times \log(y_{\text{pred}}^i) + (1 - y_i) \times (1 - \log(y_{\text{pred}}^i))) \tag{15}$$

4.3 Discussion

When users' click and wow behaviors are concerned, it is natural to take into account the article content. If we further consider articles, the problem is highly related to social recommendation. However, the main focus of this article is user behavior prediction based on user demographics, user relationships and ego network property, which is related to the problem of social influence locality, so we exclude article information to make the problem clear.

5 Experiments

We present the effectiveness of our model on users' wow and click behavior prediction of WeChat Top Stories, and also a publicly available Weibo dataset. The codes and used data in the experiments are publicly available.[2]

5.1 Experiment Setup

Datasets. We mainly evaluate our model on the collected WeChat Top Stories dataset. To further verify the generalization ability of our method, we also choose a publicly available Weibo dataset for evaluation.

For WeChat Top Stories dataset, we collect data from Oct. 1 to Dec. 31, 2019 to evaluate our proposed method. To effectively model the influence of users' friend circles on users' online behaviors, we only consider interactions with at least five friends having wowed the articles. After filtering out data, we select all positive instances, in which there are $3{,}163{,}171$ wow instances, $2{,}181{,}279$ click instances, which result in $5{,}121{,}571$ positive instances in total. We further sample a subset of negative instances randomly to keep the ratio between positive and negative instances relatively balanced. We take data from Oct. 1 to Nov. 30 for training, Dec. 1 to Dec. 19 for validation and Dec. 20 to Dec. 31 for testing, which results in $5{,}058{,}036$ training instances, $1{,}061{,}840$ validation instances, and $876{,}664$ testing instances. We keep the ratio between positive and negative instances as about $1.5 : 1$ and $1 : 1$ for wow and click datasets respectively.

Another dataset is Weibo dataset[3]. Weibo[4] is the most popular microblogging in China. The original dataset consists of the direct user following networks and tweet logs in 2012. The goal is to predict users' retweet behavior based on their local neighbors. We follow the same setup as [31]. Finally, we obtain $779{,}164$ data instances, in which 50% are used for training, 25% for validation and 25% for testing.

Comparison Methods. We compare our proposed model with the following methods.

- **Random.** We generate like/click probability uniformly in the range [0, 1) for prediction.
- **Logistic Regression (LR).** We use logistic regression (LR) to train a classification model. We define three categories of features: (1) ego users' features, including user gender, age, region, social roles (whether one is an opinion leader, a structural hole) (2) ego network features, including the number of active friends, the number of connected components (#CC) and the local clustering coefficient of the ego graph formed by active friends; (3) relation features of ego users and friends: average and sum of the common friends' ratio between ego user and each active friend.
- **Random Forest (RF) [22].** We use Random Forest to train a classification model due to its effectiveness in selecting relevant features and instances. The used features are the same as Logistic Regression.
- **xDeepFM [21].** xDeepFM is a framework based on Factorization Machine (FM), taking users' features as input. It learns high-order feature interaction with FM modules and also has DNN modules to model feature interactions implicitly. The input features are the same as LR and RF.
- **DeepInf [31].** DeepInf is a framework to learn users' latent representation for predicting social influence. It takes users' ego networks as input and uses the graph neural network to learn user representation. Here we adopt GAT to learn user embedding due to its superiority for influence prediction in the paper [31].
- **Wang et al. [43].** This method models the topological influence structure based on Weisfeiler-Lehman (WL) algorithm and learns the influence dynamics for the ego user by leveraging GAT, too. The different parts of features are concatenated to make predictions.
- **SAGPool [19].** SAGPool is a graph pooling method that uses self-attention to distinguish between nodes that should be dropped and the nodes that should be retained. The predictions are made on the smaller graphs.
- **ASAP [32].** ASAP utilizes a novel self-attention network to cluster similar nodes together in a graph. Then, the most important clusters are selected and included in the pooled graph. After each pooling step, the graph is summarized using a readout function.
- **StructPool [47].** StructPool is also a hierarchical graph pooling method, which formulates the cluster assignment problem as a structured prediction problem. It employs conditional random fields to capture the relationships among assignments of different nodes.
- **ProHENE.** Our model takes users' ego network, and user features as input. A delicate graph filter is used to transform user features in the modulated spectral domain of ego networks. Then we learn hierarchical structure embeddings of ego networks to predict ego users' behavior in an end-to-end fashion. We use $\textbf{ProHENE}_{\textbf{AA}}$ to denote using additive attention in the basic GNN modules, and use $\textbf{ProHENE}_{\textbf{DA}}$ to denote using dot attention in the basic GNN modules.

Parameter Settings. For WeChat dataset, we set the maximum number of users/nodes in the sampled ego network as 32. For the pre-trained user embeddings, We generate 64-dim embeddings using ProNE [48]. In the user feature smoothing method via graph filter, we choose the parameters in the graph filter as follows: $\mu = 0.4$, $\theta = 7$. For hierarchical graph representation learning, the number of graph coarsening steps is set as 2. In GAT encoders of hierarchical graph representation learning, we set the head number as 8 and hidden vector dimension for each head as 16. When training, the learning rate is 0.01 for wow prediction and 0.1 for click prediction. The L2 regularization

2. https://github.com/zfjsail/wechat-wow-analysis
3. http://aminer.org/Influencelocality
4. https://weibo.com

TABLE 7
Results of User Behavior Prediction.

	WeChat Wow				WeChat Click				Weibo			
Method	Prec	Rec	F1	AUC	Prec	Rec	F1	AUC	Prec	Rec	F1	AUC
Random	47.84	50.06	48.92	50.05	28.64	50.13	36.45	50.02	25.12	50.48	33.55	50.15
LR	68.08	70.08	69.06	76.73	41.71	67.01	51.41	70.07	42.97	71.37	53.64	76.38
RF [22]	69.20	65.17	67.12	76.69	39.52	74.69	51.69	70.12	40.03	73.66	51.87	75.14
xDeepFM [21]	66.23	80.96	72.85	78.25	40.88	75.09	52.94	71.61	30.20	73.90	42.88	64.38
DeepInf [31]	70.28	**81.46**	75.46	83.06	43.88	76.03	55.65	74.50	48.09	71.67	57.56	81.46
Wang et al. [43]	69.76	79.40	74.27	81.91	41.91	75.07	53.79	72.31	45.58	74.63	56.59	80.26
SAGPool [19]	81.74	75.43	78.46	86.18	46.58	79.19	58.66	77.37	43.79	73.81	54.97	78.89
ASAP [32]	71.13	79.81	75.22	83.28	44.92	76.57	56.62	75.48	46.55	70.64	56.12	79.87
StructPool [47]	67.56	79.21	72.92	79.46	40.20	78.55	53.19	71.54	30.47	72.87	42.98	61.83
$\text{ProHENE}_{\text{AA}}$	84.95	76.81	**80.67**	87.64	**46.63**	82.01	59.46	78.05	**50.09**	72.87	**59.37**	**83.08**
$\text{ProHENE}_{\text{DA}}$	**85.46**	76.30	80.62	**87.69**	46.45	**82.81**	**59.52**	**78.27**	48.70	**74.88**	59.01	82.76
w/o pre-train	74.96	78.42	76.65	84.91	45.68	75.77	57.00	76.09	47.33	74.15	57.78	81.51
w/o node feature	85.01	75.69	80.08	87.10	45.64	82.26	58.71	77.64	46.34	74.46	57.13	81.10
w/o 2nd feature	86.40	76.16	80.16	87.50	46.66	81.66	59.39	78.16	46.12	75.02	57.12	81.03
w/o smoothing	79.23	77.57	78.39	86.04	46.26	78.38	58.18	76.89	48.95	72.20	58.35	82.13

weight is 0.0005. Adagrad [8] is chosen as the optimizer. As for Weibo dataset, there are several differences as follows. Following [31], we adopt random walk with restart (RWR) to generate the sampled ego networks. The number of graph coarsening step is 1 and the learning rate is 0.05.

5.2 Overall Results

Table 7 summarizes the results of user behavior prediction. The comparison methods can be roughly divided into several categories: (1) traditional classifiers: LR and RF, (2) deep learning method by modeling feature interactions: xDeepFM, (3) the state-of-the-art user behavior prediction methods based on ego networks: DeepInf and Wang et al. and (4) hierarchical graph representation learning methods: SAGPool, ASAP and StructPool. (3) and (4) are both GNN-based methods. Generally, we observe that our model ProHENE consistently outperforms baseline methods.

For traditional classifiers, it can not achieve better prediction performance than other methods, although it leverages hand-craft user features, user relation features and network features. xDeepFM, a factorization-machine based neural network model, achieves better performance than LR and RF on WeChat dataset, which might imply that the correlation between user features is an inherent factor that impacts users' wow and click behaviors, such as the correlation between users' gender and age.

DeepInf and Wang et al. could both achieve good prediction performance on three datasets. It demonstrates that modeling pairwise user influence via graph attention is effective. However, the prediction performance of Wang et. al is inferior than DeepInf, which might indicate that sometimes local topological features could result in negative impact on user behavior prediction.

For hierarchical graph representation learning methods, SAGPool outperforms most of baselines, though still weak than ProHENE. It indicates that dropping nodes via self-attention on graphs is another effective solution to graph coarsening. Besides, ASAP performs better than SAGPool on Weibo dataset, which might imply that Weibo and WeChat datasets have different characteristics for user behavior modeling. As for StructPool, we infer that modeling cluster assignment as a structured prediction problem via conditional random fields is ineffective for our problem due to its inferior performance.

For our proposed model, its superiority could mainly attribute to hierarchical structure embedding of ego networks and the feature smoothing effect. Moreover, compared with baselines using various hand-crafted features, our method only uses user features as input. Thus, our method can model user relations and ego network structure better than hand-crafted features. Furthermore, we observe that $\text{ProHENE}_{\text{DA}}$ achieves similar performance as $\text{ProHENE}_{\text{AA}}$ and $\text{ProHENE}_{\text{DA}}$ sometimes outperforms $\text{ProHENE}_{\text{AA}}$ slightly. Here we argue that AUC is more important than F1 metric, since F1 depends on a good threshold for classification. Thus, modeling feature interactions between neighboring nodes takes effect at times.

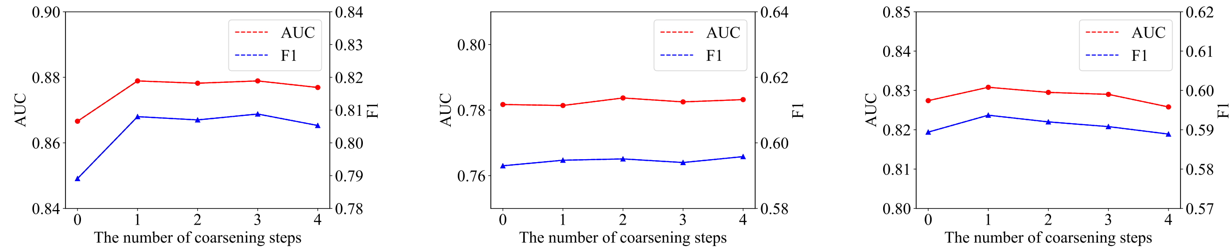
In addtion, for WeChat Top Stories dataset, we find that the performance differences between different methods for click prediction are smaller than those of wow prediction. Another observation is that, in general, the click prediction performance is much lower than wow prediction performance. This phenomenon is probably because users' click (or reading) behavior is more correlated to the articles themselves, while their wow behavior is more relevant to the social influence around them.

5.3 Ablation Study.

We study the effects of different model components for user behavior prediction.

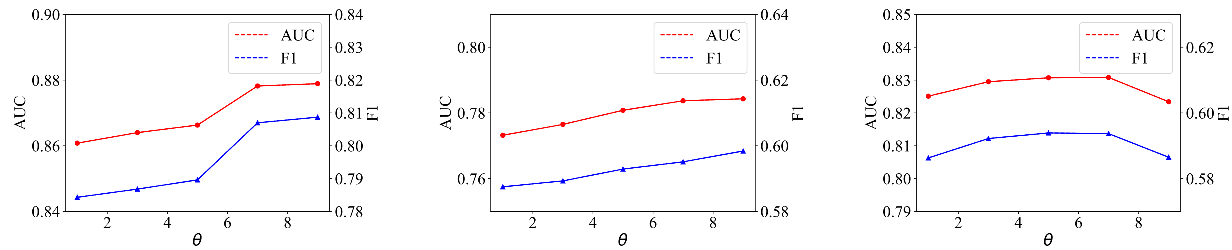
- **w/o pre-train**: Remove the pre-trained ProNE user embedding in the input features. Note that the second-order features also lack this part.
- **w/o node feature**: Remove the demographic and social social role features in the input features. Note that the second-order features also lack this part.
- **w/o 2nd feature**: Remove the second-order feature interactions of different user features in the input features.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3064233, IEEE Transactions on Knowledge and Data Engineering



(a) WeChat Wow (b) WeChat Click (c) Weibo

Fig. 11. Wow and click performance (AUC) on test dataset w.r.t. the number of pooling layers in hierarchical graph representation learning

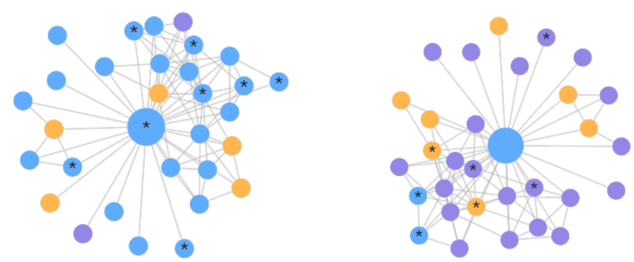


(a) WeChat Wow (b) WeChat Click (c) Weibo

Fig. 12. Prediction performance (AUC and F1) on test dataset w.r.t. θ in the graph filter of the feature smoothing step.

- **w/o smoothing**: Remove the feature smoothing step via the graph filter.

In Table 7, the bottom part summarizes the results of the ablation study. We observe that all studied components contribute to the effectiveness of our model to some degree. Among all the components, removing pre-trained embeddings and removing feature smoothing step cause larger performance drops compared to other components on WeChat dataset. In contrast, adding second-order features contributes a little to predict user behavior in WeChat Top Stories, but contributes more on Weibo dataset. Meanwhile, our model can perform well, even without input hand-crafted user features.



(a) Positive wow sample (b) Negative wow sample

Fig. 13. Visualization of cluster assignment of hierarchical graph representation learning for wow behavior prediction. (a) Positive sample: the ego user wowed the article. (b) Negative sample: the ego user didn't wow the article. Here different colors represent different cluster assignments. The largest node in each ego network is the ego user. Nodes with "*" inside mean that these users "wowed" the article.

5.4 Parameter Analysis.

The Number of Coarsening Steps in Hierarchical Graph Representation Learning. In hierarchical graph representation step, each coarsening step transforms the ego network to smaller graphs iteratively by clustering similar "nodes" together. We study whether the number of coarsening steps influences the prediction performance. Figure 11 shows the prediction performance w.r.t. the number of coarsening steps. In the experiment, we set in each iteration, the number of "nodes" becomes half of that of the last iteration. We can see that hierarchical graph representation clearly better than "flat" representation (0 pooling layer) on WeChat Wow and Weibo dataset. The prediction performance changes little in terms of the number of coarsening steps for click behavior prediction. When there are 2 coarsening steps, test AUC and F1 for click prediction is highest. 1 pooling layer is the best for Weibo dataset and WeChat wow. Perhaps although we set the coarsening ratio as 50%, GNN can automatically learn the node proximity and the appropriate number of meaningful clusters [46].

Parameters in the graph filter. We analyze how the parameters θ in the graph filter $g(\lambda) = e^{-\frac{1}{2}[(\lambda-\mu)^2-1]\theta}$ can affect the prediction performance. Here μ is trainable and we set its initial value as 0.4. Figure 12 shows the performance variations (AUC and F1) w.r.t. θ on test data. Parameter θ in g affects the peak value of modulated eigenvalue λ. We observe that when $\theta = 7$ or $\theta = 9$, AUC is the higher for wow and click prediction than other tested configurations. However, for weibo dataset, when θ varies from 1 to 9, test performance increases first then decreases.

5.5 Visualization of Hierarchical Graph Representation Learning

We visualize the hierarchical graph representation learning process to understand how it assigns nodes into different clusters and performs predictions. Figure 13(a) and 13(b) show two case studies for wow prediction, where Figure 13(a) is a positive wow instance and Figure 13(b) is a negative wow instance. The two subfigures both visualize the first coarsening step. We can observe that, although the coarsening ratio is set as 50%, which means the nodes in one ego network can be assigned to at most 16 clusters, the two examples both assign nodes into only three clusters. It shows that the learning algorithm can automatically learn the appropriate number of clusters. Moreover, for wow positive instance in Figure 13(a), all active friends (who wowed the article) are assigned to blue clusters, which may correspond to the conformity phenomenon. As for the negative wow instance in Figure 13(b), the active friends are assigned to different clusters (nodes with different colors all have nodes with "*" within). The inactivity of the ego user perhaps can be explained by the negative correlation between ego users' wow probability and structural diversity (referring to Sec. 3.3).

6 Related Work

6.1 Social Influence Analysis

Social influence has been studied and modeled widely from different viewpoints. At the macro level, the problem of influence maximization in social networks has been studied in [6], [17]. Xin et al. [35] study the indirect influence on Twitter. Micro influence like pairwise influence has been studied in [12], [34], [49]. Liu et al. [24] study the micro mechanism of influence diffusion in heterogeneous social networks and propose a probabilistic generative model. Tang et al. [38] propose Topical Affinity Propagation (TAP) to model influence on different topics. More recently, deep learning models have been proposed to model social influence. Qiu et al. [31] use Graph Attention Networks (GAT) to model social influence locality. Feng et al. [10] propose a skip-gram architecture to learn user embeddings which reflect social influence. In this work, we first analyze social influence on micro-level (e.g. pairwise influence) and local network structure, based on which we propose an effective method to model social influence.

6.2 User Feedbacks in Recommender Systems

Generally, user feedbacks in recommender system fall into two categories: explicit feedback and implicit feedback. Implicit feedback includes click, mouse movement, etc., and explicit feedback includes retweet, ratings and so on. Jawaheer et al. [16] propose a classification framework for explicit and implicit feedback based on several properties, including Cognitive Effort, User Model, Scale of Measurement, and Domain Relevance. They also compare different user feedbacks in detail on an online music recommendation service [15]. Many recommender systems try to combine different user feedbacks to improve recommendation performance. Liu et al. [25] unify explicit and implicit feedback in a matrix-factorization framework. However, Tang et al. [39] find that it is better to train different models separately for each feedback data and then combine them. In this work, we analyze both wow and click user feedbacks and discover some interesting differences between them. Then we train separate models for each feedback data.

6.3 Network Representation Learning

Recently, network representation learning at the *node level* and *graph level* has become a research hotspot. Generally, node-level representation learning approaches can be broadly categorized as (1) factorization-based approaches such as GraRep [4], NetMF [30], (2) shallow embedding approaches such as DeepWalk [29], LINE [37], HARP [5], and (3) neural network approaches [3], [20]. Recently, graph convolutional network (GCN) [18] and its multiple variants, such as GAT [42], GIN [44], have become the dominant approaches for network representation learning, thanks to the use of graph convolution that effectively fuses graph topology and node features. Furthermore, there are also some works using graph convolution architectures for graph-level representation learning, such as [19], [36], [46]. In order to generate graph representations, most works employ graph convolution encoders to generate node embeddings first, and then use some pooling or READOUT functions, such as hierarchical pooling [46] and sum operations [36]. Among hierarchical pooling methods, some [19], [32] use self-attention to select important nodes in the graph or cluster similar nodes together. Yuan et al. [47] regard node clustering problem as structured prediction problem via conditional random fields. In this work, based on users' ego networks, we first learn user/node embeddings by modulating the spectral domain of the ego networks. Then, a hierarchical graph representation method is utilized to generate graph-level embeddings. Our method is motivated by and consistent with our statistical analysis.

7 Conclusion

In this work, we use the WeChat Top Stories data to understand user preferences and wow diffusion. Our study reveals several interesting phenomena: 1) Males' click probability is higher than females', while females' wow probability is higher than males'. 2) The active rate of young generations (users of 20s) is the lowest. 3) Given the fixed number of friends wowed an article, the larger #CC (the number of connected components formed by active friends), the lower the wow probability of ego users is, but the higher the click probability is.

Based on the important discoveries, we also develop a unified model ProHENE to predict users' online behaviors. We evaluate it on the real sizable social network, and results show that the proposed model can achieve significantly better performance over several state-of-the-arts.

Acknowledgments

This work is supported by a research fund of Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

References

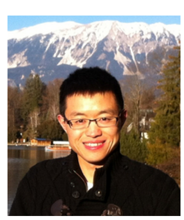
[1] L. C. Andrews. *Special functions of mathematics for engineers*, volume 49. Spie Press, 1998.

[2] R. S. Burt. Structural holes and good ideas. *American journal of sociology*, 110(2):349–399, 2004.

[3] S. Cao, W. Lu, and Q. Xu. Deep neural networks for learning graph representations. In *AAAI'16*.

[4] S. Cao, W. Lu, and Q. Xu. Grarep: Learning graph representations with global structural information. In *CIKM'15*, pages 891–900.
[5] H. Chen, B. Perozzi, Y. Hu, and S. Skiena. Harp: Hierarchical representation learning for networks. In *AAAI'18*.
[6] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD'01*, page 57–66.
[7] B. L. Douglas. The weisfeiler-lehman method and graph isomorphism testing. *arXiv preprint arXiv:1101.5211*, 2011.
[8] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159, 2011.
[9] L. Feng, Y. Hu, B. Li, H. E. Stanley, S. Havlin, and L. A. Braunstein. Competing for attention in social media under information overload conditions. *PloS one*, 10(7), 2015.
[10] S. Feng, G. Cong, A. Khan, X. Li, Y. Liu, and Y. M. Chee. Inf2vec: Latent representation model for social influence embedding. In *ICDE'18*, pages 941–952.
[11] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.
[12] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *WSDM'10*, page 241–250.
[13] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420–1443, 1978.
[14] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE TKDE*, 15(4):784–796, 2003.
[15] G. Jawaheer, M. Szomszor, and P. Kostkova. Comparison of implicit and explicit feedback from an online music recommendation service. In *Proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*, pages 47–51, 2010.
[16] G. Jawaheer, P. Weller, and P. Kostkova. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(2):1–26, 2014.
[17] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD'03*, page 137–146.
[18] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR'17*.
[19] J. Lee, I. Lee, and J. Kang. Self-attention graph pooling. In *36th International Conference on Machine Learning, ICML 2019*, pages 6661–6670. International Machine Learning Society (IMLS), 2019.
[20] R. Li, S. Wang, F. Zhu, and J. Huang. Adaptive graph convolutional neural networks. In *AAAI'18*.
[21] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *KDD'18*, pages 1754–1763.
[22] A. Liaw, M. Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.
[23] H. Lin, J. Jia, J. Qiu, Y. Zhang, G. Shen, L. Xie, J. Tang, L. Feng, and T.-S. Chua. Detecting stress based on social interactions in social networks. *IEEE TKDE*, 29(9):1820–1833, 2017.
[24] L. Liu, J. Tang, J. Han, and S. Yang. Learning influence from heterogeneous social networks. *Data Mining and Knowledge Discovery*, 25, 11 2012.
[25] N. N. Liu, E. W. Xiang, M. Zhao, and Q. Yang. Unifying explicit and implicit feedback for collaborative filtering. In *CIKM'10*, pages 1445–1448.
[26] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *KDD'12*, pages 6–14.
[27] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *ICML'16*, pages 2014–2023.
[28] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
[29] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *KDD'14*, pages 701–710.
[30] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM'18*, pages 459–467.
[31] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang. Deepinf: Social influence prediction with deep learning. In *KDD'18*, pages 2110–2119.
[32] E. Ranjan, S. Sanyal, and P. P. Talukdar. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In *AAAI*, pages 5470–5477, 2020.
[33] E. M. Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.
[34] K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *KES'08*, page 67–75.
[35] X. Shuai, Y. Ding, J. Busemeyer, S. Chen, Y. Sun, and J. Tang. Modeling indirect influence on twitter. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(4):20–36, 2012.
[36] F.-Y. Sun, J. Hoffmann, and J. Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR'20*.
[37] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *WWW'15*, pages 1067–1077.
[38] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD'09*, pages 807–816.
[39] L. Tang, B. Long, B.-C. Chen, and D. Agarwal. An empirical study on recommendation with multiple types of feedback. In *KDD'16*, pages 283–292.
[40] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.
[41] J. Ugander, L. Backstrom, C. Marlow, and J. Kleinberg. Structural diversity in social contagion. *PNAS*, 109(16):5962–5966, 2012.
[42] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *ICLR'18*.
[43] 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.
[44] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *ICLR'19*.
[45] Y. Yang, J. Tang, C. Leung, Y. Sun, Q. Chen, J. Li, and Q. Yang. Rain: Social role-aware information diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.
[46] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec. Hierarchical graph representation learning with differentiable pooling. In *NeurIPS'18*, pages 4800–4810.
[47] H. Yuan and S. Ji. Structpool: Structured graph pooling via conditional random fields. In *International Conference on Learning Representations*, 2020.
[48] J. Zhang, Y. Dong, Y. Wang, J. Tang, and M. Ding. Prone: fast and scalable network representation learning. In *IJCAI'19*, pages 4278–4284.
[49] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. Social influence locality for modeling retweeting behaviors. In *IJCAI'13*, pages 2761–2767.

Fanjin Zhang is a PhD candidate in the Department of Computer Science and Technology, Tsinghua University. She got her bachelor degree from the Department of Computer Science and Technology, Nanjing University. Her research interests include data mining and social network.



Jie Tang received the PhD degree from Tsinghua University. He is a full professor in the Department of Computer Science and Technology, Tsinghua University. His main research interests include data mining, social network, and machine learning. He has published over 200 research papers in top international journals and conferences.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3064233, IEEE Transactions on Knowledge and Data Engineering

Xueyi Liu is an undergraduate with the department of Computer Science and Technology, Tsinghua University. Her main research interests include graph representation learning, graph neural network and reasoning.

Zhenyu Hou is an undergraduate with the department of Computer Science and Technology, Tsinghua University. His main research interests include graph representation learning and reasoning.

Yuxiao Dong received his Ph.D. in Computer Science from University of Notre Dame in 2017. He is an applied scientist at Microsoft Research Redmond. His research focuses on social networks, data mining, and computational social science, with an emphasis on applying computational models to addressing problems in large-scale networked systems.

Jing Zhang received the master and PhD degree from the Department of Computer Science and Technology, Tsinghua University. She is an assistant professor in Information School, Renmin University of China. Her research interests include social network mining and deep learning.

Xiao Liu is a senior undergraduate student with the Department of Computer Science and Technology, Tsinghua University. His main research interests include data mining, machine learning and knowledge graph. He has published papers on KDD.

Ruobing Xie is a researcher of WeChat, Tencent. He got his BEng degree in 2014 and his master degree in 2017 from the Department of Computer Science and Technology, Tsinghua University. His research interests are natural language processing and recommender system. He has published over 15 papers in international journals and conferences including IJCAI, AAAI, ACL and EMNLP.

Kai Zhuang received his master degree from the Department of Computer Science and Technology, Sichuan University, China, in 2013. He is a researcher in WeChat Search Application Department, Tencent Inc., China.

Xu Zhang received his master degree in College of Computer Science and Engineering, Northeastern University in 2008. He is the team leader of WeChat User Profile, Tencent. His research interests include machine learning and its applications, such as social network, user interest model and user behavior research.

Leyu Lin received his master degree in Institute of Computing Technology, Chinese Academy of Sciences, in 2008. He is currently the deputy director of WeChat Search Team, Tencent. His research interests include machine learning and its applications, such as search system, recommendation system and computational advertising.

Philip S. Yu received the PhD degree in electrical engineering from Stanford University. He is a distinguished professor in computer science with the University of Illinois, Chicago, and is the Wexler chair in Information Technology. His research interests include big data, data mining, data stream, database, and privacy. He was the editor-in-chief of the IEEE Transactions on Knowledge and Data Engineering and the ACM Transactions on Knowledge Discovery from Data. He received the ACM SIGKDD 2016 Innovation Award, a Research Contributions Award from the IEEE International Conference on Data Mining (2003), and a Technical Achievement Award fromthe IEEEComputer Society (2013). He is a fellow of the IEEE and ACM.