Social Behavior Analysis in Exclusive Enterprise Social Networks by FastHAND

YANG YANG*, Defense Innovation Institute, Chinese Academy of Military Science, Beijing, China
FEIFEI WANG†, Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing, China
ENQIANG ZHU, Institute of Computing Science and Technology, Guangzhou University, Guangzhou, China
FEI JIANG, School of Electronics Engineering and Computer Science, Peking University, Beijing, China
WEN YAO‡, Defense Innovation Institute, Chinese Academy of Military Science, Beijing, China

There is an emerging trend in the Chinese automobile industries that automakers are introducing exclusive enterprise social networks (EESNs) to expand sales and provide after-sale services. The traditional online social networks (OSNs) and enterprise social networks (ESNs), such as Twitter and Yammer, are ingeniously designed to facilitate unregulated communications among equal individuals. However, users in EESNs are naturally social stratified, consisting of both enterprise staffs and customers. In addition, the motivation to operate EESNs can be quite complicated, including providing customer services and facilitating communication among enterprise staffs. As a result, the social behaviors in EESNs can be quite different from those in OSNs and ESNs. In this work, we aim to analyze the social behaviors in EESNs. We consider the Chinese car manufacturer NIO as a typical example of EESNs and provide the following contributions. First, we formulate the social behavior analysis in EESNs as a link prediction problem in heterogeneous social networks. Second, to analyze this link prediction problem, we derive plentiful user features and build multiple meta-path graphs for EESNs. Third, we develop a novel Fast (H)eterogeneous graph (A)ttention (N)etwork algorithm for (D)irected graphs (FastHAND) to predict directed social links among users in EESNs. This algorithm introduces feature group attention at the node-level and uses an edge sampling algorithm over directed meta-path graphs to reduce the computation cost. By conducting various experiments on the NIO community data, we demonstrate the predictive power of our proposed FastHAND method. The experimental results also verify our intuitions about social affinity propagation in EESNs.

CCS Concepts: • **Networks** → **Online social networks**; • **Computer systems organization** → **Neural networks**; • **Theory of computation** → *Sparsification and spanners*; • **Human-centered computing** → **Social network analysis**; **Social networking sites**.

Additional Key Words and Phrases: Heterogeneous Social Network, Directed Graphs, Graph Attention Neural Network, Link Prediction, Graph Spectral Sparsification

*Also with Intelligent Game and Decision Laboratory, Chinese Academy of Military Science.
†Corresponding Author.
‡Corresponding Author.

Authors' addresses: Yang Yang, bigyangy@gmail.com, Defense Innovation Institute, Chinese Academy of Military Science, No.53, Fengtai East Street, Beijing, China, 100071; Feifei Wang, feifei.wang@ruc.edu.cn, Center for Applied Statistics and School of Statistics, Renmin University of China, No.59 Zhongguancun Street, 100872, Beijing, China; Enqiang Zhu, zhuenqiang@gzhu.edu.cn, Institute of Computing Science and Technology, Guangzhou University, Guangzhou, China, 510006; Fei Jiang, fei.jiang1989@pku.edu.cn, School of Electronics Engineering and Computer Science, Peking University, No.5, Yiheyuan Road, Haidian District, Beijing, China, 100871; Wen Yao, wendy0782@126.com, Defense Innovation Institute, Chinese Academy of Military Science, No.53, Fengtai East Street, 100071, Beijing, China.

ACM 1556-4681/2024/2-ART
https://doi.org/10.1145/3646552

1 INTRODUCTION

With the rapid development of Internet, many traditional business models have been changed dramatically. For example, third-party online platforms (e.g., Yelp, Airbnb) spring out, creating a communication bridge for merchants and customers. In the meanwhile, more and more enterprises have built their own communities to provide better service for customers. One typical example is the Chinese automobile industry. Supported by "The Internet Plus" plan launched by the Chinese government in 2014, many new energy vehicle enterprises have been established, such as NIO, Zeekr, XPeng and LI. These enterprises maintain an enterprise-level community for their customers or potential customers to keep in close touch and to provide life long sales and after-sale services for automobile owners. These requirements breed a novel form of social networks, which we call *exclusive enterprise social networks* (EESNs).

The social services provided by EESNs are similar with traditional social networks. For example, users are free to post, comment, thumb-up, and follow each other. Users can also take private chats, join topic groups, take part in online and offline community activities, etc. However, the motivations of building traditional social networks and EESNs are completely different. Traditional social networks are designed to satisfy the intrinsic requirements of users, such as socialization, corporation and communication. On the contrary, EESNs are primarily designed to provide regulated customer services, such as advertisement marketing, pre-sales consulting and after-sale services. Users in EESNs are inundated with propaganda news pushed by the community system, which includes enterprise advertising posts and intentionally selected user posts. Consequently, there exist two major differences between EESNs and their forerunners. First, as illustrated in Figure 1, personal news and therefore user influence are suppressed to propagate through social relations, which is contradictory to the basic logic of peer influence in traditional online social networks (OSNs) and enterprise social networks (ESNs) [6, 12, 25]. Second, there exists two groups of users: a small but important group constituted by enterprise staffs, who also act as the community managers and customer service providers; a large group of normal users, whose initial motivation to register EESNs is to absorb merchandise-centered advertisements and instructions. However in traditional OSNs, there are only normal users while in ESNs there are only enterprise staffs [1, 33].

Given its uniqueness, we aim to investigate social behaviors in this novel social network. Understanding the social behaviors of EESNs has several important business implications. First, EESNs offer a unique platform for enterprises to engage with their customers. Understanding how social interactions occur within EESNs can help businesses improve customer engagement and retention. By analyzing social behaviors, enterprises can tailor their communication strategies, offer better customer support, and create more effective marketing campaigns. Second, EESNs provide a channel for enterprises to promote their products and services directly to a user base interested in their industry or offerings. By studying user behaviors and preferences, businesses can refine their targeting strategies and optimize their marketing efforts to reach potential customers more effectively. In summary, a better understanding of EESNs is essential for businesses to leverage the unique opportunities they offer for customer engagement, targeted marketing, community building, and staying competitive in the digital age. It enables enterprises to make informed decisions and optimize their strategies for better business outcomes.

To investigate the social behaviors of EESNs, we raise two specific questions in this work:

1) How do social influences propagate across an EESN that lead to the acquaintances and social link formation among its users?
2) Do enterprise staffs and normal users behave similarly or differently in making friends?

To answer these two questions, we take the NIO community (https://app-intl.nio.com/) as a representative. The NIO is a luxury electric automaker in China and has gained twice the value of BMW's stock market value at its highest point. Accordingly, the NIO community, with its claimed 1.6 million registrations, is one of the largest and most active EESNs in China. Thus in this work, we consider NIO as an example to analyze social behaviors of EESN users and answer the two questions proposed above.

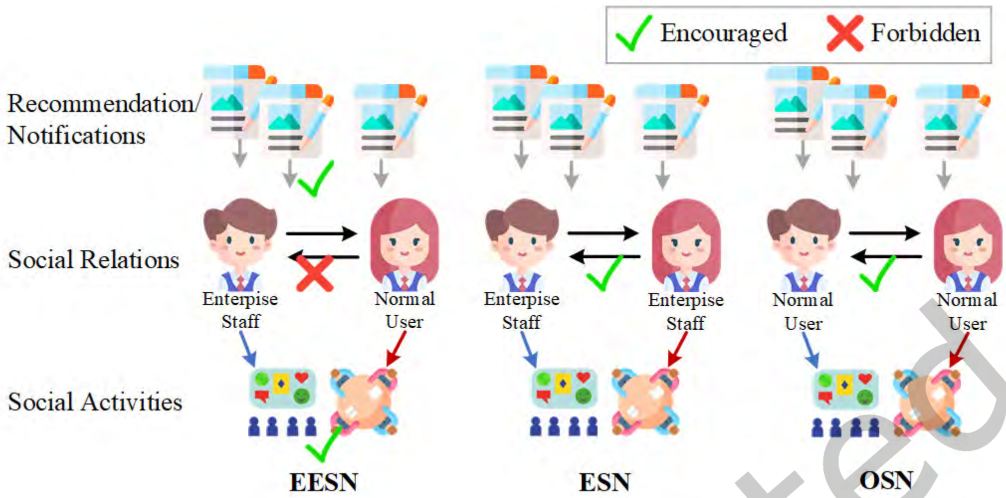

Fig. 1. There are two groups of users in EESNs. News are directly pushed to all users through front page recommendation or as system notifications; social activities such as topic groups chatting and offline group meetings are encouraged and promoted. However, posts sharing through social relations and user timeline views are forbidden and discouraged.

We formulate social behavior analysis as a link prediction problem [14]. The NIO can be modeled as a heterogeneous social network, where we regard users, posts, comments, community activities, user topic groups and other entities as *heterogeneous nodes*, while the observed relations among different types of pairs of nodes as *heterogeneous edges*, such as user making posts and user joining online community activities. For users, we extract comprehensive features from their portals to characterize their personalities and build multiple meta-paths that associate pairs of users to reveal their regulated social interactions in NIO. Then we proceed with a hypothesis that social behaviors of users are heavily influenced through those regulated social tunnels. In addition, users get to know each other through those tunnels and make personal decisions about making friends. To this end, we apply a self-attention mechanism over the meta-paths. Also note that the NIO network is huge, directed, and discrete. Thus we develop a fast (H)eterogeneous graph (A)ttention (N)etwork algorithm for (D)irected graphs, which we call FastHAND for short. The FastHAND algorithm utilizes spectrum preserving edge sampling to compress the number of directed meta-paths to improve computational efficiency. It also applies a heterogeneous graph attention network to allow user features flow along meta-paths. As a result, FastHAND is feasible for graphs of large scale, compared with traditional graph convolutional network methods.

We then apply FastHAND on the NIO network dataset and find both extracted user features and meta-paths are informative and useful in predicting social link formation. Based on these properties, the FastHAND algorism not only validates our hypothesis about regulated social tunnels in NIO, but also exhibits robustness against many other competitive methods in different task settings. Through feature ablation study, we also find the same feature might have different influences to different types of social links, which corresponds to our second question about social behaviors. The major contributions of our work can be summarized as follows:

- Based on observations of a novel ESN NIO, we propose the FastHAND algorithm that formulates social behavior analysis as link prediction over directed and heterogeneous social graphs.

- To release high computation burden of self-attentions in GAT layers of FastHAND, we introduce effective resistance sampling algorithm to directed graphs and theoretically prove that its spectral sparsification preserves features in every GAT layer.
- To solve data unquantifiable and incompleteness problems of user features, we apply a novel penalized PageRank algorithm to rank job titles of NIO staffs and a modified CAMLP[27] algorithm to predict geo-locations that are left blank by users.
- Experimental evaluations demonstrate the effectiveness of FastHAND in social link prediction and our intuitions about social behavior in NIO.

The rest of the paper is organized as follows. In Section 2, we briefly review related works. In Section 3, we introduce the NIO dataset. In Section 4, we formulate the link prediction problem for EESNs. In Sections 5-7, we respectively introduce feature extraction, meta-path construction and the FastHAND algorithm. In Section 8, we show the experimental results on the NIO network. Finally, we conclude this paper in Section 9.

2 RELATED WORKS

2.1 Graph Representation for Link Prediction

Link prediction, a crucial technique for understanding network dynamics, has witnessed significant attention over the past decade. Recently, there has been a surge of interest in employing Graph Neural Networks (GNNs) within an encoder-decoder framework for link prediction. In this framework, the GNN-based encoder learns node representations, while the decoder predicts link existence probabilities. A notable approach is the graph autoencoder[9], which leverages Graph Convolutional Networks (GCNs) for message passing and predicts links using pairs of node embeddings. However, it's worth noting that GNNs often face challenges in capturing structural information, and they may underperform simple heuristics, such as common neighbor and Katz index. To address this limitation, SEAL [35] proposes extracting a subgraph around the target link and learning a mapping function that translates the subgraph pattern into link existence probabilities. On the other hand, Neo-GNN [32] integrates node features with structural features derived from the adjacency matrix to enhance performance.

In the context of directed links, a prominent method involves learning two latent embeddings (source and target) for each node to capture asymmetric interrelationships. The HOPE framework [15] assumes that nodes with more and shorter paths between them should have more similar source and target embeddings. It introduces a source-target embedding paradigm to preserve the feasibility of asymmetric transitivity in directed graphs. The APP method [38] employs a random walk with a restart approach to generate positive node pairs in the forward direction. These pairs are then used to train a source-target embedding model, which learns to predict the direction of the link between the source and target nodes. Salha et al. [20] draw inspiration from Newton's theory of universal gravitation to propose a decoder scheme for directed link prediction. This scheme utilizes a source-target embedding paradigm to effectively reconstruct asymmetric relationships from vector space node embeddings. In a similar vein, Jiao et al. [7] focus on predicting social and anchor links across multiple aligned social networks. They introduce a hierarchical graph attention mechanism and a source-target embedding paradigm to learn representations of nodes by aggregating information from both intra-network neighbors and inter-network partners.

2.2 Graph Sparsification for GNN Acceleration

Graph sparsification, a foundational technique for expediting graph algorithms, strategically trims task-irrelevant edges based on optimization objectives. Recent advances extend this to preprocessing input graphs for Graph Neural Networks (GNNs), vastly improving computational efficiency and memory access during training. Two key strategies have emerged: Heuristic Sparsification and Learnable Sparsification Modules[11]. In Heuristic Sparsification, DropEdge [18] introduces randomness by selectively removing edges in each training epoch, countering over-smoothing during deep GNN training. FastGAT [21] employs a resistance-based spectral graph sparsification, substantially reducing the number of attention coefficients in a Graph Attention Network (GAT) model. Methods in Learnable Sparsification Modules normally treat graph sparsification as an optimization problem. SGCN [10] and SGAT [29] both define indicator functions and formulate graph sparsification as a minimization problem with differentiable loss of GNN and non-differentiable indicator function. NeuralSparse [37] leverages a deep neural network to parameterize the sparsification process and learn a distribution of edge existence informed by downstream task feedback. GAUG [36] utilizes a neural edge predictor to identify and remove noisy edges, resulting in a sparser graph that facilitates GNN in discerning crucial features and node relations. As FastGAT significantly compresses edges while preserving the graph spectrum with theoretical guarantee, this paper extends the algorithm to handle directed graphs, alleviating the computational load of computing attention coefficients in EESN social networks.

3 THE NIO DATASET

3.1 Data Description

The NIO dataset is collected from the NIO APP, which supports the NIO community. Figure 2 illustrates the interface and some featured pages in the NIO APP. The dataset is collected in two stages. The first stage was conducted in December, 2020 and we focused on collecting social relations to build the social graph. The second stage was conducted in January, 2021 and we iterated over the graph to thoroughly collect user profiles and contents.

Social relations. To build the social graph of NIO community, we started with its CEO and implemented a broad-first search to iteratively crawl all users through directed social relations with no more than depth 5. Through this method, we collected 13,383,656 directed social links, among them 19.60% were reciprocal links, and observed 921,954 users, including 2,377 NIO staffs and 919,577 non-staff users, which we call normal users.

User profile. We collected profile for every user. The metadata of user profile includes user headshot, registration time, gender, geo-location (city and province of the user), following count (the number of followees), follower count (the number of followers), liked count (the number of "thumbed ups" from other users), and participation of user groups. It also includes the "NIO staff" tag, indicating whether a user is a NIO staff or not, and an "identity" tag, which labels a user as car owner, co-owner, intentional owner or potential owner. The "NIO staff" and "identity" tags are in parallel. In other words, a user can simultaneously be a NIO staff and car owner. Besides, each NIO staff is also associated with a unique "job title", which identifies his/her detailed job position. Based on our observation, there are 250 NIO staffs with 204 different job titles working in the headquarter, including the company founder, president, 11 vice presidents and 33 department directors. The rest 1,929 NIO staffs work in regional distributed branches, with 83 distinct job titles. From the profile, we also observe that, users participate in online community groups such as local car clubs, and offline activities such as "back to school" charity rally. In this regard, we observe 7,776 users participating in 2,814 offline activities and 15,036 users joining 185 user chat groups.

User content. The user contents include user posts and comments, from which we are able to infer their topical preferences and build heterogeneous social relations. In total, there are 28,489 users who have produced 238,459 original posts, and 125,578 users who have made 1,672,622 comments under those posts. Since users are encouraged to use hashtags in posts to identify their attendance of online community activities, we observe there are 6,195 users joining 1,012 online activities.

3.2 The Uniqueness of NIO Network

Major differences between NIO and traditional social networks are their diversified user identities and unique information propagation tunnels.

User identities. Normally, users in OSNs are usually ordinary individuals and users in ESNs are all enterprise staffs, while the NIO social network is a hybrid. Recall there exists two groups of users in NIO, including one minor but important group consisting NIO staffs and a majority group consisting normal customers.

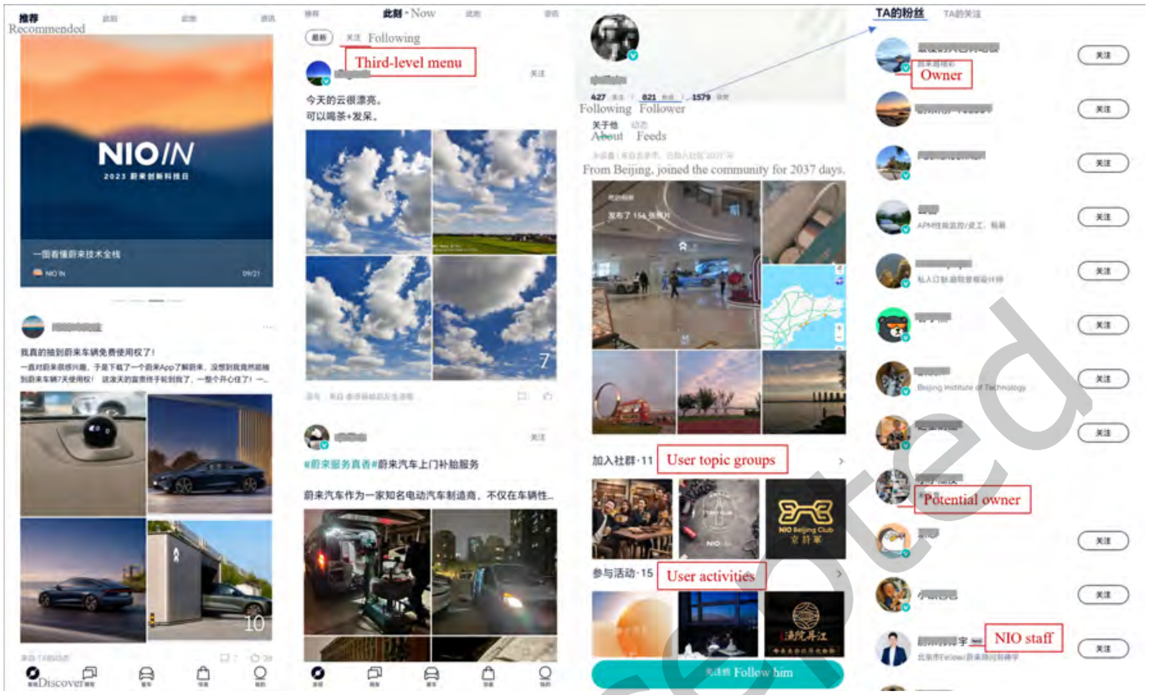

Fig. 2. NIO Interface and User Profile with Connections. **Column 1**: The front page of the NIO app showcases a curated selection of recommended user posts and advertisements. **Column 2**: Within the "Discovery" section, friends' timeline views are accessible through the third-level menu under the "Discover > Now > Following" hierarchy. **Column 3**: A typical user profile is displayed, offering options to navigate to the user's followees, followers, published content, joined user topic groups, and community activities. **Column 4**: The user's followers are presented, along with their associated social identities, including "car owner", "potential owner", and "NIO staff".

Information propagation. Generally, in OSNs, users socialize to maintain their offline friendships[4, 8] and make new friends; while in ESNs, enterprise staffs intend to communicate and collaborate with colleagues[23]. During these processes, social link plays an important role in associating existing friends and therefore is the major tunnel that allows user interests and affinities to propagate. However this is not the case in NIO. In NIO community, free and unregulated information flows are implicitly prevented. For example, the “timeline” button listing users' history publications is hidden deep in the third-level menu, and the “share to friends” function is also not provided. As a result, a user can neither get easy access to his/her friends daily updates, nor can he/she directly share interests through existing associations. On the contrary, users are pushed with marketing materials, such as selected user publications in headline and official news pushed as system notifications.

However, the NIO community innovatively creates some additional social tunnels. One is community activities that held both online and offline for users and staffs to meet and remain active. In offline activities, users are gathered by NIO staffs to engage in face-to-face meetings, such as seeds sharing and charity gatherings. In online activities, users are encouraged to make well-designed posts to share about designated topics, such as car driving experiences. Besides, user specialists are officially supported to organize special business user groups, such as preschool teachers and lawyers. Through these social tunnels, users in the NIO community socialize and get familiar with each other, potentially driving the formation of friendships. Therefore the NIO network is significantly different from "social link based" influence propagations in OSNs and ESNs.

The additional social tunnels provided by NIO community are vital for information flow. Users who initially register for marketing purposes are not familiar with each other but fortunately get exposed to each other through different additional social tunnels. During this process, they may watch one's headshot and gender to get the first impression, note the "NIO staff" and "identity" tags to judge one's authorities, and view publications to know one's taste. Then they make decisions about whether to follow the user or not, which are based on propagated user tastes with different user focuses. Therefore we aim to investigate the influence of additional social tunnels to the formations of social relations.

4 PROBLEM FORMULATION

From the description of NIO network, we find it is quite different with traditional social networks. Therefore, it is of great importance to study the properties of this novel social network. To this end, we formulate the investigation of NIO as a link prediction problem. Below, we first introduce some important definitions and then give the outline to solve the link prediction problem. The used symbols are summarized in Table 1.

Definition 1. Heterogeneous information network (HIN). Following [22], we represent NIO as a heterogeneous network $G = (V, E)$, where $V = \cup_i V_i$ are the sets of different kinds of nodes and $E = \cup_j E_j$ are different types of links in the network. Figure 3 illustrates the network template T_G. As shown, the node set $V = U \cup A \cup P \cup C \cup G$ consists of (U)sers, community (A)ctivities, (P)osts, (C)omments and (T)opic groups. Let $U = U_e \cup U_r$ define the set of all users, where U_e defines the node set of enterprise staffs and $U_r = U_o \cup U_{co} \cup U_{io} \cup U_{po}$ is the node set of regular users, including (e)nterprise staffs, (o)wners, (co)-owners, intentional owners (io), and potential owners (po). Each user u_i from U is associated with a vector of feature s_i. Let $A = A_{on} \cup A_{off}$ denote the activity collections containing (on)line activities and (off)line activities. The edge set $E = E_{U \to U} \cup E_{U \to A_{on}} \cup E_{U \to A_{off}} \cup E_{U \to P} \cup E_{U \to C} \cup E_{C \to P} \cup E_{C \to C}$ contains social links among users, user activity attendances from users to online and offline community activities, user posting behaviors from users to posts, user commenting relations from users to comments, comments following posts and comments following other comments.

Based on the HIN network template that associates different types of nodes in EESNs, we proceed to define meta-path graphs to depict different social tunnels among pairs of users.

Definition 2. Meta-path graph. A meta-path p is a path defined on the network template T_G. Denote a meta-path by $U \xrightarrow{E_1} V_1 \xrightarrow{E_2} V_2 \cdots V_{l-1} \xrightarrow{E_l} U$, which describes a specific relation among pairs of users. Then a meta-path graph $\Phi = \cup\phi$ is the union of all corresponding meta-paths ϕ. For example, the $UA_{on}U$ graph is defined as $\Phi_{UA_{on}U} = \cup\phi_{UA_{on}U}$, where $\phi_{UA_{on}U} = U \xrightarrow{E_{U \to A_{on}}} A_{on} \xleftarrow{E_{U \to A_{on}}} U$ is a reflection of user common participants of online activities.

We therefore define social graph to induce the definition of our prediction task.

Definition 3. Social graph and link prediction. The social graph of NIO is defined as $G' = (U, E'_{U \to U})$, where nodes are NIO users and $E'_{U \to U}$ are observed social links among them, which is a proper subset of $E_{U \to U}$. Then the link prediction problem is to make binary classification of candidate social links in $E' = U \times U \backslash E'_{U \to U}$.

5 FEATURE EXTRACTION

In conventional practice, structural features employed for link prediction are derived from social graphs, reflecting users' social preferences. In this study, we delve into the formation of social links using user profile information and meta-paths. However, before extracting user profile features, the completion of missing information within these profiles is necessary. Notably, approximately 8.3% of NIO staff members have omitted their job titles, and 22.38% of users have omitted geo-locations. To address this, we propose a novel algorithm for quantifying job level ranks, which assigns a job level score to each NIO staff member with missing job title information. Additionally,

Table 1. The summary of used symbols and their descriptions in problem formulation.

Symbol	Description
$G = (V, E)$	Heterogeneous network.
T_G	Network template of G, illustrating different types of nodes and their relationships
$V = \{v_i \mid 1 \leq i \leq N\}$	Node set in G.
$U = \{u_k \mid 1 \leq k \leq K\}$	Set of user nodes, including regular users and enterprise staffs. Subset of V.
U_e	Set of enterprise staffs, subset of U.
U_r	Set of regular users, subset of U.
U_o	Set of car owners, subset of U_r.
U_{co}	Set of car co-owners, subset of U_r.
U_{io}	Set of car intentional owners, subset of U_r.
U_{po}	Set of car potential owners, subset of U_r.
A	Set of community activities, subset of V.
A_{on}	Set of online activities, subset of A.
A_{off}	Set of offline activities, subset of A.
P	Set of user posts, subset of V.
C	Set of user comments, subset of V.
T	Set of user topic groups, subset of V.
E	Edge set in G.
$\rightarrow, \leftarrow$	Directed links from nodes in source set to target set.
$E_{U \rightarrow U}$	Social links among users, subset of E.
$E'_{U \rightarrow U}$	Observed social links among users, subset of E.
$E_{U \rightarrow A_{on}}$	Users attend online activities, subset of E.
$E_{U \rightarrow A_{off}}$	Users attend offline activities, subset of E.
$E_{U \rightarrow P}$	Users publish posts, subset of E.
$E_{U \rightarrow C}$	Users make comments, subset of E.
$E_{C \rightarrow P}$	Comments following posts, subset of E.
$E_{C \rightarrow C}$	Comments following each other, subset of E.
Φ	A meta-path graph.
ϕ	A meta-path from Φ.
$G' = (U, E'_{U \rightarrow U})$	Social graph consisting of user set U and observed social links $E'_{U \rightarrow U}$.
$E' = U \times U \backslash E'_{U \rightarrow U}$	Candidate social links for prediction.
$\Gamma(u_k)$	Neighboring user set of u_k in G.
$\Gamma_{in}(u_k)$	In-neighboring user (follower) set of u_k in G.
Δ_{in}	Average number of in-neighboring users for all users in G.
$\Gamma_{out}(u_k)$	Out-neighboring user (followee) set of u_k in G.
Δ_{out}	Average number of out-neighboring users for all users in G.
$\Gamma_{inter}(u_k)$	Friend user set of user u_k, $\Gamma_{inter}(u_k) = \Gamma_{in}(u_k) \cap \Gamma_{out}(u_k)$.
$\Gamma'_{in}(u_k)$	In-neighboring user list of user u_k in G.
$\Gamma'_{out}(u_k)$	Out-neighboring user list of user u_k in G.
$\Gamma'_{inter}(u_k)$	Friend user intersection of $\Gamma'_{in}(u_k)$ and $\Gamma'_{out}(u_k)$.
$\Gamma'(u_k)$	Neighboring user list of user u_k in G.
$\Gamma^{up}_{in,staff}(u_k)$	Set of in-neighboring users who are in higher job rank level than u_k.
$\Gamma^{down}_{in,staff}(u_k)$	Set of in-neighboring users who are in lower job rank levels than u_k.
$\Gamma^{up}_{in,own}(u_k)$	Set of in-neighboring users who are in higher car-owner level than u_k.
$\Gamma^{down}_{in,own}(u_k)$	Set of in-neighboring users who are in lower car-owner levels than u_k.
$\Gamma^{up}_{out,staff}(u_k)$	Set of out-neighboring users who are in higher job rank level than u_k.
$\Gamma^{down}_{out,staff}(u_k)$	Set of out-neighboring users who are in lower job rank levels than u_k.
$\Gamma^{up}_{out,own}(u_k)$	Set of out-neighboring users who are in higher car-owner level than u_k.
$\Gamma^{down}_{out,own}(u_k)$	Set of out-neighboring users who are in lower car-owner levels than u_k.
s_k	Feature vector for u_k.
#	The count value of.
@	Refer to a user in posts or comments.

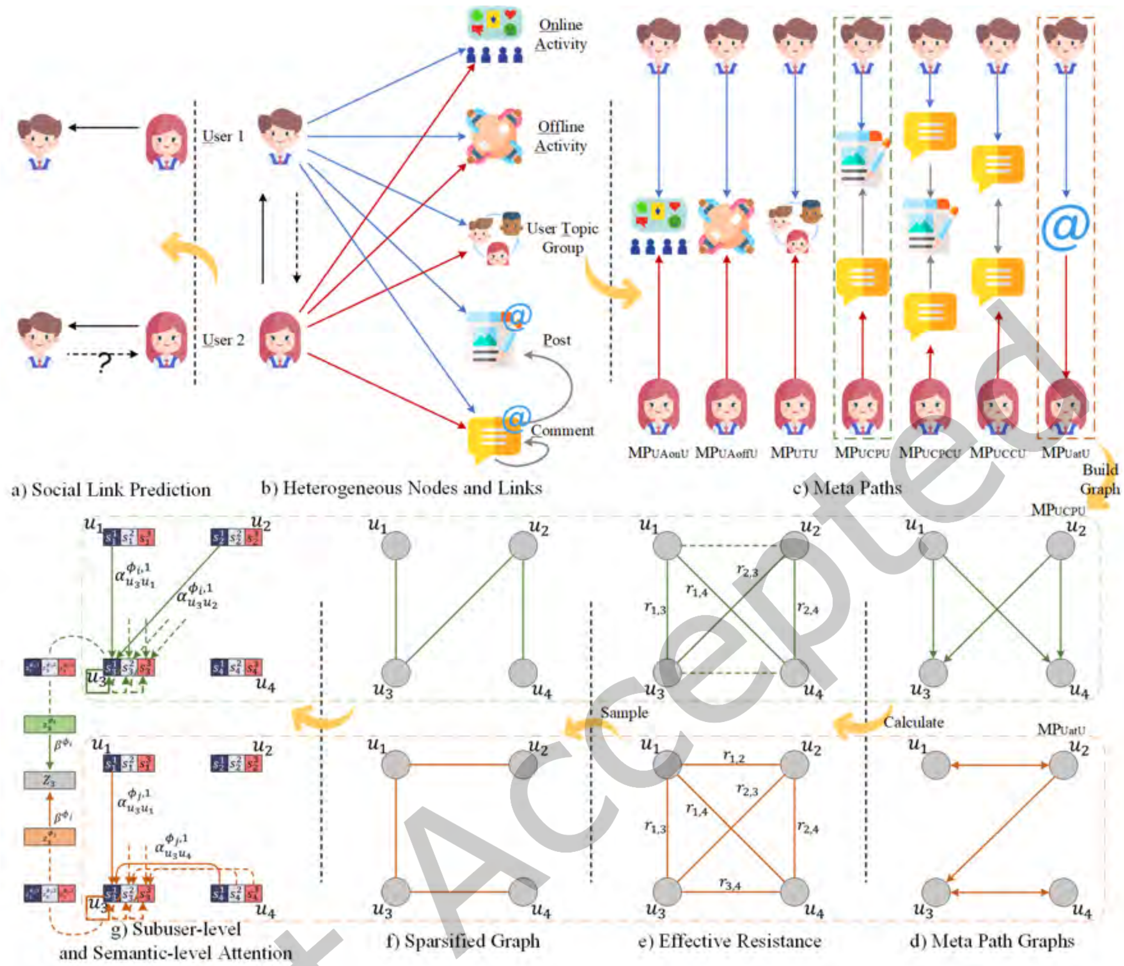

Fig. 3. The overall framework of social behaviour analysis. a) The social link prediction task. b) The network template T_G of NIO. c) meta-paths defined over T_G. d) Two illustrative directed meta-path graphs. e) Calculation of effective resistance. f) Graph sparsification based on effective resistance sampling. g) Subuser-level attention and semantic-level attention.

we tackle missing geo-location entries by applying an adapted version of the CAMLP label propagation algorithm [27]. The details of the two algorithms and corresponding results are presented in Appendices A and B. With the completed information, we proceed to capture user characteristics from different aspects, categorizing them into three groups: (1) *basic statistics* from user profile, (2) *social behavior statistics* from friend lists, and (3) *user generated contents* from user posts and comments. Below, we offer a comprehensive introduction to the detailed features extracted from each group, and a summarized overview is provided in Table 2.

Table 2. Summary of user features, which are classified into three groups: the basic statics (BS), the social behavior statistics (SBS), and the user generated contents (UGC).

Group	Subgroup	Feature	Type	Description
Basic Statics (BS)	Social identity	BS_{own}	Ord	User being car owner, co-owner, intentional owner or potential owner
		BS_{staff}	Bin	User being NIO staff or not
		BS_{gender}	Bin	User being female or male
		$BS_{jobrank}$	Real	Job rank score
	Social relations and activities	$BS_{\#follower}$	Int	Number of followers
		$BS_{\#followee}$	Int	Number of followees
		$BS_{\#group}$	Int	Number of joined user groups
		$BS_{\#online}$	Int	Number of joined online activities
		$BS_{\#offline}$	Int	Number of joined offline activities
		$\frac{BS_{\#followee}}{BS_{\#follower}}$	Real	Social attractiveness
		$\frac{BS_{\#follower}}{BS_{\#day}}$	Real	Number of followers gained per day
		$\frac{BS_{\#followee}}{BS_{\#day}}$	Real	Number of followees gained per day
		$\frac{BS_{\#group}}{BS_{\#day}}$	Real	Number of user groups joined per day
		$\frac{BS_{\#online}}{BS_{\#day}}$	Real	Number of online activities joined per day
		$\frac{BS_{\#offline}}{BS_{\#day}}$	Real	Number of offline activities joined per day
	Geo-locations	BS_{geo}	Ord	Ordinal encoding of users in mainland China
	User headshot	$BS_{headshot}$	Vector	10-dim topic distribution of user headshot caption
	Link begging	BS_{beggar}	Bin	Binary indicator of a user being "followship beggar" or not
Social Behavior Statistics (SBS)	Social activeness and awareness	$\frac{\lvert\Gamma_{out}(u_k)\rvert}{\Delta_{out}}$	Real	Social activeness
		$\frac{\lvert\Gamma_{in}(u_k)\rvert}{\Delta_{in}}$	Real	Social awareness
	Reciprocal rate and social acceptance	$\frac{\lvert\Gamma_{inter}(u_k)\rvert}{\lvert\Gamma_{in}(u_k)\rvert}$	Real	Incoming reciprocal rate
		$\frac{\lvert\Gamma_{inter}(u_k)\rvert}{\lvert\Gamma_{out}(u_k)\rvert}$	Real	Outgoing reciprocal rate
		$\frac{\lvert\Gamma_{in}(u_k)\rvert}{\lvert\Gamma_{out}(u_k)\rvert}$	Real	Social acceptance
	Consistency of reciprocal orders	$\frac{Lev(\Gamma'^{in}_{inter}(u_k),\Gamma'^{out}_{inter}(u_k))}{\lvert\Gamma'^{in}_{inter}(u_k)\rvert}$	Real	Consistency of reciprocal order
	Social transcendence rate	$\frac{\lvert\Gamma^{up}_{in,staff}(u_k)\rvert}{\lvert\Gamma_{in}(u_k)\rvert}$	Real	Upward NIO-staff social transcendence rate of followers
		$\frac{\lvert\Gamma^{down}_{in,staff}(u_k)\rvert}{\lvert\Gamma_{in}(u_k)\rvert}$	Real	Downward NIO-staff social transcendence rate of followers
		$\frac{\lvert\Gamma^{up}_{out,staff}(u_k)\rvert}{\lvert\Gamma_{out}(u_k)\rvert}$	Real	Upward NIO-staff social transcendence rate of followees
		$\frac{\lvert\Gamma^{down}_{out,staff}(u_k)\rvert}{\lvert\Gamma_{out}(u_k)\rvert}$	Real	Downward NIO-staff social transcendence rate of followees
		$\frac{\lvert\Gamma^{up}_{in,own}(u_k)\rvert}{\lvert\Gamma_{in}(u_k)\rvert}$	Real	Upward car-owner social transcendence rate of followers
		$\frac{\lvert\Gamma^{down}_{in,own}(u_k)\rvert}{\lvert\Gamma_{in}(u_k)\rvert}$	Real	Downward car-owner social transcendence rate of followers
		$\frac{\lvert\Gamma^{up}_{out,own}(u_k)\rvert}{\lvert\Gamma_{out}(u_k)\rvert}$	Real	Upward car-owner social transcendence rate of followees
		$\frac{\lvert\Gamma^{down}_{out,own}(u_k)\rvert}{\lvert\Gamma_{out}(u_k)\rvert}$	Real	Downward car-owner social transcendence rate of followees
User Generated Contents (UGC)	User posting behaviors	$UGC_{\#post}$	Int	Number of posts
		$UGC_{\#comment}$	Int	Number of comments
		$UGC_{\#thumb}$	Int	Number of thumbed-ups from others
		$\frac{UGC_{\#post}}{BS_{\#day}}$	Real	Posting rate
		$\frac{UGC_{\#thumb}}{UGC_{\#post}}$	Real	Average thumbed-ups gained by each post
		$\frac{UGC_{\#thumb}}{UGC_{\#post}\times BS_{\#day}}$	Real	Average thumbed-ups gained by each post per day
		$UGC_{\#Pcomment}$	Int	Number of comments gained by posts
		$\frac{UGC_{\#Pcomment}}{BS_{\#day}}$	Real	Number of comments gained by posts per day
		$\frac{UGC_{\#Pcomment}}{UGC_{\#post}\times BS_{\#day}}$	Real	Number of comments gained by each post per day
		$UGC_{\#Rcomment}$	Int	Number of comments gained by comments
		$\frac{UGC_{\#Rcomment}}{UGC_{\#comment}}$	Real	Number of comments gained by each comment
		$\frac{UGC_{\#Rcomment}}{UGC_{\#comment}\times BS_{\#day}}$	Real	Number of comments gained by each comment per day
		$UGC_{\#at}$	Int	Number of referred to by "@" from others
		$\frac{UGC_{\#at}}{BS_{\#day}}$	Real	Number of referred to by "@" per day
	User topics	UGC_{topic}	Vector	20-dim topic distribution of post content
	Social promotion	$UGC_{\#promote}$	Int	The count of promoted posts
		$\frac{UGC_{\#promote}}{UGC_{\#post}}$	Real	Promoted times per post
		$\frac{UGC_{\#promote}}{BS_{\#day}}$	Real	Promoted posts per day

5.1 Basic Statistics

Within the *Basic Statistics (BS)* group, we collect features from user profiles that serve as fundamental indicators of users' social status, engagement, attractiveness, and popularity.

Social identities. We employ the notation BS_{own} to denote a user's car ownership status, which is organized into four levels: "car owner", "co-owner", "intentional owner", and "potential owner". These levels are structured in descending order of importance, with "car owner" being the highest rank, followed by "co-owner", "intentional owner", and finally "potential owner". For gender distinction, we use BS_{gender} to identify users as female or male. Additionally, we introduce BS_{staff} to identify users who are staff members. To measure the importance of staff roles, we introduce a job rank inference algorithm outlined in Appendix A. The algorithm assigns a penalized PageRank value to each staff member, denoted as $BS_{jobrank}$.

Social relations and activities. We compute several counts summarizing users' social links and activities. These include the count value of his/her followers and followees ($BS_{\#follower}$, $BS_{\#followee}$), the number of joined user groups, online and offline activities ($BS_{\#group}$, $BS_{\#online}$, and $BS_{\#offline}$). To account for differences between long-standing users and newcomers, we normalize all count features by the user's registration time ($BS_{\#day}$).

Geo-locations. Based on the geo-location analysis in Appendix B, we uncover explicit patterns in social relations across various geo-locations. We employ ordinal encoding to represent users' geo-locations, represented as BS_{geo}.

In addition to commonly-used features, we introduce two novel types of features extracted from user headshots and comments.

User headshots. Drawing inspiration from the notion that users with visually appealing headshots are more likely to make favorable impressions [30], we embed each headshot into a 10-dimensional vector for link formation prediction. Initially, a pre-trained natural image captioning algorithm [26] generates a concise sentence for each headshot. Subsequently, we employ the Bi-term topic model tailored for short texts [28] to compress each sentence into a distribution over 10 topics. The 10-dimensional topic distribution of user headshots is denoted as $BS_{headshot}$.

Link begging. We recognize a distinct category of user comments that request followships, typified by phrases like "I have followed you, and you're welcomed to follow back." Users displaying this behavior are termed "followship beggars." We employ BS_{beggar} to denote whether a user falls into this category or not.

5.2 Social Behavior Statistics

We have identified intriguing patterns in the dynamics of friendship formation within the NIO context. For instance, certain users intentionally follow a substantial number of individuals to ensure the reciprocation of social links, which we coin as "link farmers." Conversely, some users actively engage with their followers, and we refer to them as "link responders." Furthermore, we recognize the role of social stratification hierarchy in influencing the establishment of links and integrate this insight into both the NIO-staff and car-owner tag systems.

To encapsulate these distinct user behaviors, we embark on our exploration by first establishing fundamental definitions and subsequently introducing a suite of features termed *social behavior statistics (SBS)*. For a given user u_k, let $\Gamma'_{in}(u_k)$ be the time-ordered list of his/her followers and $\Gamma_{in}(u_k)$ be the set of followers without orders. Then the average number of followers across the community is calculated as $\Delta_{in} = \frac{1}{K}\sum_k |\Gamma_{in}(u_k)|$, where K is the total number of users. Similarly, we define $\Gamma'_{out}(u_k)$, $\Gamma_{out}(u_k)$ and $\Delta_{out} = \frac{1}{K}\sum_k |\Gamma_{out}(u_k)|$ to encapsulate the information about his/her followees. Lastly, we define the user's friend set as $\Gamma_{inter}(u_k) = \Gamma_{in}(u_k) \cap \Gamma_{out}(u_k)$, which is the intersection of his/her follower and followee sets.

Social activeness and awareness. The social activeness and awareness of user u_k are defined as $\frac{|\Gamma_{out}(u_k)|}{\Delta_{out}}$ and $\frac{|\Gamma_{in}(u_k)|}{\Delta_{in}}$. A large value of social activeness means the user is exceptionally active in socialization. Similarly, a large value of social awareness means the user is better accepted than others.

Incoming/Outgoing reciprocal rate and social acceptance. The incoming and outgoing reciprocal rates are defined as $\frac{|\Gamma_{inter}(u_k)|}{|\Gamma_{in}(u_k)|}$ and $\frac{|\Gamma_{inter}(u_k)|}{|\Gamma_{out}(u_k)|}$. When incoming and outgoing reciprocal rates of a user are both high, he/she is prone to be a “mutual link aider". Besides, the social acceptance of user u_k is defined as $\frac{|\Gamma_{in}(u_k)|}{|\Gamma_{out}(u_k)|}$. With higher attentions gained from the community (i.e., larger $|\Gamma_{in}(u_k)|$) and less efforts paid to socialize (i.e., smaller $|\Gamma_{out}(u_k)|$) with others, user u_k is prone to gain higher social acceptance.

Consistency of reciprocal orders. Let $\Gamma'^{in}_{inter}(u_k)$ be the reordered set of $\Gamma_{inter}(u_k)$ following the time-order of $\Gamma'_{in}(u_k)$, and $\Gamma'^{out}_{inter}(u_k)$ be the reordered set of $\Gamma_{inter}(u_k)$ following $\Gamma'_{out}(u_k)$. We then utilize the *Levenshtein distance* to calculate their differences $Lev(\Gamma'^{in}_{inter}(u_k), \Gamma'^{out}_{inter}(u_k))$. Then the consistency of reciprocal orders for u_k is defined as $SBS^{consist}_{recip} = \frac{Lev(\Gamma'^{in}_{inter}(u_k), \Gamma'^{out}_{inter}(u_k))}{|\Gamma'^{in}_{inter}(u_k)|}$. Social hierarchy and stratification is a natural phenomenon in social networks. For example, the organizational chart is a reflection of users' hierarchical management levels in ESNs. Social links that associate users in different social stratifications are called social transcendence links. In recent studies, social hierarchies are found influential to the formation of social links[1, 33]. Thus we evaluate user characteristics with Upward/Downward social transcendence rate metrics.

Upward/Downward social transcendence rate. The NIO community exhibits social stratification through its NIO-staff and car-owner identities. As previously outlined, each user is assigned a $BS_{jobrank}$ score to identify the importance of their job roles. This allows us to define $|\Gamma^{up}_{in,staff}(u_k)|$ as the count of followers who have higher job rank scores than u_k, and $|\Gamma^{down}_{in,staff}(u_k)|$ as the count of followers who have lower job rank scores than u_k. Then the upward and downward NIO-staff social transcendence rates of u_k are calculated as $\frac{|\Gamma^{up}_{in,staff}(u_k)|}{|\Gamma_{in}(u_k)|}$ and $\frac{|\Gamma^{down}_{in,staff}(u_k)|}{|\Gamma_{in}(u_k)|}$. Regarding car-owner status, we adhere to the structured levels of BS_{own} to define $|\Gamma^{up}_{in,own}(u_k)|$ as the count of followers who have higher car-owner status than u_k and $|\Gamma^{down}_{in,own}(u_k)|$ to be the number of followers who have lower car-owner status than u_k. Consequently, the upward and downward car-owner social transcendence rates of u_k are determined as $\frac{|\Gamma^{up}_{in,own}(u_k)|}{|\Gamma_{in}(u_k)|}$ and $\frac{|\Gamma^{down}_{in,own}(u_k)|}{|\Gamma_{in}(u_k)|}$. As for out-neighboring users of u_k, the corresponding upward/downward social transcendence rates are defined as $\frac{|\Gamma^{up}_{out,staff}(u_k)|}{|\Gamma_{out}(u_k)|}$, $\frac{|\Gamma^{down}_{out,staff}(u_k)|}{|\Gamma_{out}(u_k)|}$, $\frac{|\Gamma^{up}_{out,own}(u_k)|}{|\Gamma_{out}(u_k)|}$ and $\frac{|\Gamma^{down}_{out,own}(u_k)|}{|\Gamma_{out}(u_k)|}$. For the sake of brevity and notational clarity in the following paragraphs, we use SBS^{staff}_{trans} to denote the 4-dim NIO-staff social transcendence rates and SBS^{own}_{trans} to denote car-owner social transcendence rates.

The proposed metrics can be effectively employed in conjunction with other metrics to conduct a comprehensive assessment of a user's social characteristics. To illustrate, “link farmers” exhibit notable traits, including high social activeness $\frac{|\Gamma_{out}(u_k)|}{\Delta_{out}}$, high social awareness $\frac{|\Gamma_{in}(u_k)|}{\Delta_{in}}$ and relatively diminished social acceptance $\frac{|\Gamma_{in}(u_k)|}{|\Gamma_{out}(u_k)|}$. “Link responders” are characterized by high incoming reciprocal rate $\frac{|\Gamma_{inter}(u_k)|}{|\Gamma_{in}(u_k)|}$ and high consistency of reciprocal orders $SBS^{consist}_{recip}$. Besides, when considering a user with a higher degree of social awareness ($\frac{|\Gamma_{inter}(u_k)|}{|\Gamma_{out}(u_k)|}$), higher consistency in reciprocal orders ($SBS^{consist}_{recip}$), but lower social activeness ($\frac{|\Gamma_{out}(u_k)|}{\Delta_{out}}$), it suggests that the user garners substantial attention from the community while judiciously cultivating only a select number of relationships. This distinct pattern underscores the user's influence and implies a potential higher job level or placement within elevated social strata.

5.3 User Generated Contents

User posting behaviors are demonstrated to be influential to the formation of social links. In the NIO context, users have the capability to craft meticulously designed posts, featuring a blend of text, images, and videos. We subsequently extract features aimed at capturing the essence of user posting behaviors and user generated content (UGC).

User posting behaviours. For each user, define $UGC_{\#post}$, $UGC_{\#comment}$, $UGC_{\#thumb}$, and $UGC_{\#at}$ to be the number of his/her posts, comments, received thumbed-ups and referred to by “@” from others. Define $UGC_{\#Pcomment}$ and $UGC_{\#Rcomment}$ to be the number of comments gained by his/her posts and comments. Then we use $UGC_{\#post}$ and

$\frac{UGC_{\#post}}{BS_{\#day}}$ to evaluate the user's dynamic characteristics of posting behaviors. To evaluate the dynamic characteristics of a user's opinions accepted by others, we use four features $UGC_{\#thumb}$, $\frac{UGC_{\#thumb}}{BS_{\#day}}$, $\frac{UGC_{\#thumb}}{UGC_{\#post}}$, and $\frac{UGC_{\#thumb}}{UGC_{\#post} \times BS_{\#day}}$. To evaluate the dynamics of user's communication with other people, we use the following features $UGC_{\#Pcomment}$, $\frac{UGC_{\#Pcomment}}{BS_{\#day}}$, $\frac{UGC_{\#Pcomment}}{UGC_{\#post} \times BS_{\#day}}$, $UGC_{\#Rcomment}$, $\frac{UGC_{\#Rcomment}}{UGC_{\#post}}$, $\frac{UGC_{\#Rcomment}}{UGC_{\#post} \times BS_{\#day}}$, $UGC_{\#at}$, and $\frac{UGC_{\#at}}{BS_{\#day}}$.

User topics. We extract topical features from user publications. For each user, we concatenate all texts from a user's posts into a longer document and apply the latent Dirichlet Allocation (LDA) model [3] to infer a 20-dim topic distribution for the document, which is denoted by UGC_{topic}.

Social promotion. Recall from the uniqueness of NIO network that users are pushed with promoted official news and selected user posts. To capture the influence of social promotion to social relations, we proceed to evaluate a user in two aspects: to what extent a user has been promoted by NIO and what is his/her preference to promoted users. For the first aspect, define $UGC_{\#promote}$ to be the count of promoted posts of a user. Then his/her social promotion rate can be defined as $\frac{UGC_{\#promote}}{UGC_{\#post}}$ and $\frac{UGC_{\#promote}}{BS_{\#day}}$. For the second aspect, we average the three social promotion metrics over his/her followers and followees, separately.

6 META-PATH CONSTRUCTION

Users in NIO socialize and get familiar with each other through explicit social interactions, such as officially organized online activities, offline gatherings, and posting and commenting behaviors. We then build meta-paths to depict the social channels that allow user features to flow from user to user. In summary, we conduct three groups of meta-paths. They are, respectively: (1) *common activities* meta-paths based on the observations of user participation in online and offline community activities, (2) *common user group* meta-paths that associate users who have joined the same user groups, and (3) *posting behaviors* meta-paths that link users who interact with each other through posting and comments.

6.1 Common Activities

In the NIO dataset, there are 2,814 offline activities for 7,776 users. The offline activities provide good chance for online community users to build friendships through face-to-face socialization. However, participants in online activities are less exposed to each other and thus they have less chance to get familiar with each other. Therefore, the offline activities are more discriminative than online activities to capture social affinity of participants. We build meta-paths for both offline and online activities. Specifically, for the community activities between user u_i and u_j, the meta-paths are defined as:

- $MP_{UA_{on}U} : U \rightarrow A_{on} \leftarrow U$, common online activity,
- $MP_{UA_{off}U} : U \rightarrow A_{off} \leftarrow U$, common offline activity,

where U denotes user nodes, A_{on} and A_{off} denote online activities and offline activities. Let $|\Gamma_{A_{on}}(u_k)|$ denote the number of online activities that user u_k joins in and $|A_{on}|$ denote the number of online activities. Then the common online activity membership affinity between u_i and u_j is defined as:

$$P(MP_{UA_{on}U}(u_i, u_j)) = \frac{1}{|\Gamma_{A_{on}}(u_i)|} \frac{1}{|\Gamma_{A_{on}}(u_j)|} \frac{|\Gamma_{A_{on}}(u_i) \cap \Gamma_{A_{on}}(u_j)|}{|A_{on}|}.$$

Similarly, the common offline activity membership affinity between u_i and u_j is defined as:

$$P(MP_{UA_{off}U}(u_i, u_j)) = \frac{1}{|\Gamma_{A_{off}}(u_i)|} \frac{1}{|\Gamma_{A_{off}}(u_j)|} \frac{|\Gamma_{A_{off}}(u_i) \cap \Gamma_{A_{off}}(u_j)|}{|A_{off}|}.$$

6.2 Common User Group

Following the definition of common activity meta-paths, we further define common topic group meta-path to depict social channels provided by user topic groups for users to socialize. Specifically, the meta-path is defined as

- $MP_{UTU}: U \rightarrow T \leftarrow U$,

where U denotes user nodes and T denotes topic groups. Let $|\Gamma_T(u_k)|$ denote the number of topic groups that user u_k joins in and $|T|$ denote the number of topic groups. Then the probability of user u_i and user u_j sharing topic groups can be defined as:

$$P(MP_{UTU}(u_i, u_j)) = \frac{1}{|\Gamma_T(u_i)|} \frac{1}{|\Gamma_T(u_j)|} \frac{|\Gamma_T(u_i) \cap \Gamma_T(u_j)|}{|T|}.$$

6.3 Posting Behaviors

In the NIO community, users are free to post and comment to other posts. These behaviors contribute to different social interactions among users. For instance, users who comment to a post may share similar interest with the author of the post. Users who comment under the same post may be motivated by similar reasons and share similar interest. Besides, users who discuss with each other under the same post or “refer to” each other also indicate they share similar interest. To reflect this idea, we construct four meta-paths to capture the social interactions between u_i and u_j:

- $MP_{UCPU}: U \rightarrow C \rightarrow P \leftarrow U$,
- $MP_{UCCU}: U \rightarrow C \rightarrow C \leftarrow U$,
- $MP_{UCPCU}: U \rightarrow C \rightarrow P \leftarrow C \leftarrow U$,
- $MP_{UatU}: U \rightarrow @ \rightarrow U$,

where U denotes user nodes, C denotes user comments, P denotes user posts and @ denotes “refer to” tags. As a result, the meta-path MP_{UCPU} depicts the social behavior that u_i comments to posts written by u_j. Let $|C_{u_i}|$ be the number of comments written by user u_i, $|P_{u_j}|$ be the number of posts user u_j submitted to the community, and $I_1(\cdot, \cdot)$ represent an indicator function that determines whether the m-th comment $C_{u_i,m}$ directly follows the n-th post $P_{u_j,n}$ or not. Then the probability of user u_i commenting to user u_j's posts can be computed as follows:

$$P(MP_{UCPU}(u_i, u_j)) = \frac{1}{|C_{u_i}|} \frac{1}{|P_{u_j}|} \sum_{m=1}^{|C_{u_i}|} \sum_{n=1}^{|P_{u_j}|} I_1(C_{u_i,m}, P_{u_j,n}).$$

Similarly, the meta-path MP_{UCCU} is built to depict two users commenting with each other under some posts. The formation probability of MP_{UCCU} from u_i to u_j is:

$$P(MP_{UCCU}(u_i, u_j)) = \frac{1}{|C_{u_i}|} \frac{1}{|C_{u_j}|} \sum_{m=1}^{|C_{u_i}|} \sum_{n=1}^{|C_{u_j}|} I_2(C_{u_i,m}, C_{u_j,n}),$$

where $I_2(C_{u_i,m}, C_{u_j,n})$ indicates if comment $C_{u_i,m}$ is a direct follower of comment $C_{u_j,n}$. Another way for users to get familiar with each other is that, they both leave comments under the same post, indicating shared interests. We therefore build the meta-path MP_{UCPCU} to depict this situation with the probability given below:

$$P(MP_{UCPUC}(u_i, u_j)) = \frac{1}{|C_{u_i}|} \frac{1}{|C_{u_j}|} \sum_{m=1}^{|C_{u_i}|} \sum_{n=1}^{|C_{u_j}|} I_3(C_{u_i,m}, C_{u_j,n}),$$

where $I_3(C_{u_i,m}, C_{u_j,n})$ indicates whether comment $c_{u_{i_m}}$ and comment $c_{u_{j_n}}$ appear under the same post. For meta-path MP_{UatU}, let $|@_{u_i,u_j}|$ denote the number of times that user u_i “refer to” user u_j and $|@|$ denote the total number of "refer to" tags. Then the probability of user u_i referring to user u_j can be defined as $P_{(MP_{UatU})} = \frac{|@_{u_i,u_j}|}{|@|}$. Overall, the $MP_{UA_{on}U}$, $MP_{UA_{off}U}$ and MP_{UTU} are undirected meta-paths and MP_{UCCU}, MP_{UCPU} and MP_{UatU} are directed meta-paths, that All weights derived in the above process are normalized and then used to define the incidence matrix in the edge resistance sampling algorithm (see Appendix C for details). Finally, we summarize the meta-paths of the three groups in Table 3.

Table 3. Summary of meta-paths, which are classified into three groups: the common activities, the common user group, and the posting behaviors.

Meta-path	Description
Common Activities	
$MP_{UA_{on}U}$	Two users join in a same online activity.
$MP_{UA_{off}U}$	Two users attend a same offline gathering.
Common User Group	
MP_{UTU}	Two users join in a same topic group.
Posting Behaviors	
MP_{UCPU}	User u_i write comments to posts created by user u_j.
MP_{UCPCU}	Users u_i and u_j write comments to a same post.
MP_{UCCU}	Users u_i and u_j comment to each other under a post.
MP_{UatU}	Users u_i mention u_j in a post or comment.

7 THE FASTHAND METHOD

This section introduces the FastHAND algorithm for social relation analysis and link prediction. It involves three layers attention mechanisms, i.e., subuser-level attention, user-level attention, and semantic-level attention, which can be used to sufficiently utilize multi-modal features extracted from NIO users. To release the heavy computation for self-attention coefficients between connected users in different meta-path graphs, we develop an edge sampling algorithm for directed graph in FastHAND.

7.1 Attention Mechanism

Subuser-level Attention. In most GNN applications, node features are commonly treated as a unified set. While in EESNs, the multifaceted features of a user can be intuitively categorized into distinct groups, each capturing different facets of the user's characteristics. As a result, we refer to these feature groups as sub-user level features and introduce the subuser-level attention to investigate their significance. Specifically, assume feature vector for user u_i is s_i, where $s_i \in \mathbb{R}^F$. The feature vector can be divided into M groups, i.e., $s_i = \sqcup_m s_{i_m}$, which we call sub-user level features. Then for user u_i and user u_j who are connected in an arbitrary meta-path graph Φ, the self-attention coefficient from user u_i to user u_j over meta-path ϕ can be defined as:

$$e^{\phi,m}_{u_i,u_j} = a^m(W^m s^m_i, W^m s^m_j; \Phi),$$

given meta-path graph Φ, a^m denotes the shared subuser-level attention mechanism and $W^{\phi,m}$ denotes the shared subuser level weight matrix applied to the mth feature group. The intuition here is that, users recognizing each other through the same meta-path may behave similarly. For instance, when users chat in online topic groups (MP_{UTU}), they would undoubtedly look at the others' headshots to see the gender, age, tastes, etc. For users who meet each other in offline community activities ($MP_{UA_{off}U}$), they may behave more interested of others' UGCs for deeper understandings.

We proceed to inject structural information into the FastHAND algorithm via masked attention. Specifically, we only calculate $e_{ij}^{\phi,m}$ for user $u_j \in$ given meta-path graph Φ and user u_i, we only calculate his/her attentions to user $u_j \in \Gamma_{out,\phi}(u_j)$, where $\Gamma_{out,\phi}(u_j)$ denotes the out-neighboring set of user u_i (including him/herself) over meta-path graph Φ. Then we have

$$e_{u_i,u_j}^{\phi,m} = \sigma(a^{\Phi,m^T}[W^m s_i^m || W^m s_j^m]),$$

where σ denotes the activation function, $||$ denotes the concatenation operation, and $a^{\Phi,m}$ is the subuser-level feature attention vector for the mth feature group shared across the meta-path Φ. To normalize the importance between meta-path Φ for all node pairs, we adjust the weight coefficients as follows:

$$\alpha_{u_i,u_j}^{\phi,m} = softmax(e_{u_i,u_j}^{\phi,m}) = \frac{\exp(e_{u_i,u_j}^{\phi,m})}{\sum_{u_k \in \Gamma_{out,\phi}(u_i)} \exp(e_{u_i,u_k}^{\phi,m})}.$$

Finally, the meta-path based embeddings of the m-th feature group can be aggregated by user u_i's all neighbors:

$$z_i^{\Phi,m} = \sigma(\sum_{u_j \in \Gamma_{out,\phi}(u_i)} \alpha_{u_i,u_j}^{\Phi,m} W^m s_j^m).$$

User-level attention. After obtaining the subuser-level embeddings for all feature groups, the user-level embedding is an ordered concatenation of its feature group embeddings. Specifically, for user u_i, we have:

$$z_i^{\Phi} = \mathop{||}_{m=1}^{M} z_i^{\Phi,m}.$$

To stabilize the learning process of self-attention, we also extend user-level attention to multi-head attentions. Specifically, we repeat the user-level attention for K times and then average the learned embeddings as a semantic-specific embedding. It leads to the following output feature representation:

$$\widetilde{z}_i^{\Phi} = \frac{1}{K}\sum_k z_{i_k}^{\Phi}.$$

Then for user u_i, given his/her associated meta-path set $\{\Phi_1, \ldots, \Phi_P\}$, we are able to obtain P groups of user-level embeddings, denoted as $\{\widetilde{z}_i^{\Phi_1}, \widetilde{z}_i^{\Phi_2}, \ldots, \widetilde{z}_i^{\Phi_P}\}$.

Semantic-level attention. After obtaining user-level attentions in all meta-paths, we proceed to apply the user embeddings from all social channels revealed by meta-paths. Denoting the importance of each meta-path by w^{Φ_p}, we have:

$$w^{\Phi_p} = \frac{1}{|U|}\sum_{u_i \in U} q^T\, tanh(W z_i^{\Phi_p} + b),$$

where W is the weight matrix, b is the bias vector, q is the semantic-level attention vector, and U is user nodes set. Then we normalize the obtained weights for all meta-paths through the softmax function:

$$\beta^{\Phi_p} = \frac{\exp(w^{\Phi_p})}{\sum_{p=1}^{P}\exp(w^{\Phi_p})}.$$

With the learned weights as coefficients, the final semantic-level embeddings Z_i for user u_i are obtained by fusing user-level embeddings through meta-paths:

$$Z_i = \sum_{p=1}^{P} \beta^{\Phi_p} \widetilde{z}_i^{\Phi_p}.$$

7.2 Link Prediction Using FastHAND

We model the link prediction problem in EESNs as a binary classification task. To predict whether there exists a directed social link from user u_i to user u_j, we establish a multiple-layer perceptron (MLP) classifier to map user embeddings into social relations. Specifically, we use the ordered concatenation of user embeddings $Z_{u_i}||Z_{u_j}$ to denote social affinity from user i to user j. Then the MLP takes it as input and utilizes a hidden layer with softmax activation to map the concatenation into binary labels.

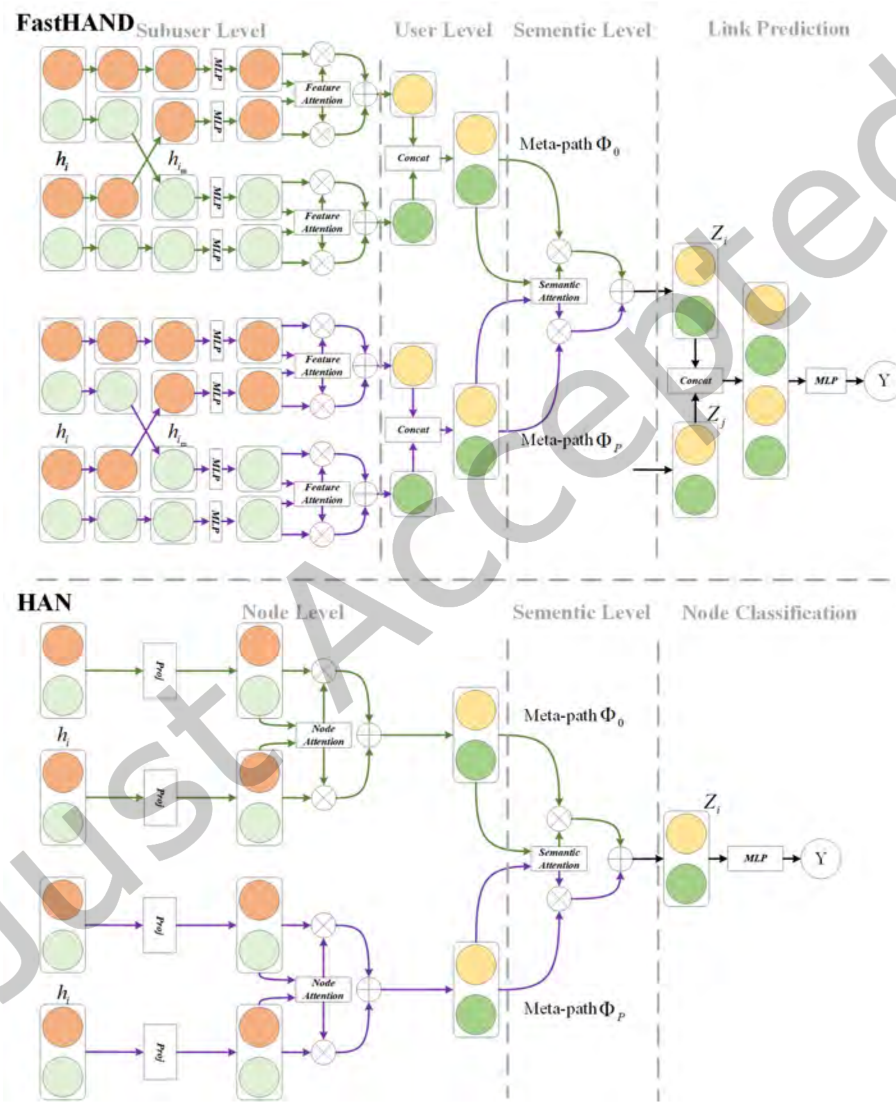

Fig. 4. Frameworks of FastHAND and HAN. The HAN learns attention weights for each paired nodes, while FastHAND separates node features into several feature groups and learns attention weights for each feature group of the paired nodes.

For better illustration, we compare the frameworks of FastHAND and HAN [13] in Figure 4. From this figure, we see clearly the differences between FastHAND and HAN. Specifically, one difference is the existence of an additional subuser-level attention layer in FastHAND, which learns self-attention coefficients between feature groups. However in HAN, self-attention weights are learned in user-level. Another difference is that, FastHAND concatenates node embeddings to denote social relations, while HAN directly maps node embeddings into class labels.

7.3 Spectral Sparsification by Resistance Sampling

To compress the requirement of computing attention coefficients, we follow [31] to expand the sampling algorithm designed for undirected graphs [21] to accommodate directed graphs. By proposing a novel sampling function $EdgeSampleD(\cdot)$, the extension serves to diminish the computational demands associated with attention coefficient calculations in directed graphs.

We proceed by giving some concepts and notations. Let graph G consist of the triple (V, E, W), where $V = \{1, 2, ..., N\}$ is the set of nodes, $E \subseteq V \times V$ is the set of edges and $W \in \mathbb{R}^{N\times N}$ is a weighted matrix with non-negative entries $w_{i,j}$. A path in G is a sequence of nodes such that each node is a neighbor of the previous one. A connection in G between nodes k and j consists of two paths, one starting at k and the other at j. They both terminate at the same node. A connection subgraph between nodes k and j in the graph is a maximal connected subgraph of G in which every node and edge form part of a connection between nodes k and j. That is, a connection subgraph is formed from the union of connections between nodes k and j, and the addition of any other connections would make the subgraph disconnected. If only one connection subgraph exists in G between nodes k and j, it is referred to as the connection subgraph and is denoted by $C_G(k, j)$. In addtion, let I_N denote the identity matrix in $\mathbb{R}^{N\times N}$, $\mathbf{1}_N$ denote the vector in $\mathbb{R}^N$ containing 1 in every entry, and $\Pi := I_N - (1/N)\mathbf{1}_N\mathbf{1}_N^T$ denote the orthogonal projection matrix onto the subspace of $\mathbb{R}^N$ perpendicular to $\mathbf{1}_N$. Use $\mathbf{1}_N^\perp$ to denote the subspace. Then let $Q \in \mathbb{R}^N$ be a matrix whose rows form an orthogonal basis for $\mathbf{1}_N^\perp$, and let $\boldsymbol{\delta}_N^k$ be the kth N-dimensional standard basis vector that contains a zero in every position except the kth position, which is a 1. Then the effective resistance between any two nodes in the graph can be defined in *Definition 4*.

Definition 4. Let G be a connected digraph with N nodes and L be its associated Laplacian matrix. Then the effective resistance between nodes k and j is defined as:

$$r_{k,j} = (\boldsymbol{\delta}_N^k - \boldsymbol{\delta}_N^j)^T X(\boldsymbol{\delta}_N^k - \boldsymbol{\delta}_N^j) = x_{k,k} + x_{j,j} - 2x_{k,j},$$

where $x_{k,j}$ represents the element located in the kth row and jth column of matrix X and

$$\begin{aligned} X &= 2Q^T\Sigma Q, && (1)\\ \overline{L}\Sigma + \Sigma\overline{L}^T &= I_{N-1}. && (2)\\ \overline{L} &= QLQ^T. && (3) \end{aligned}$$

Given triple $G = (V, E, W)$, it is easy to calculate its Laplacian matrix L. By constructing Q from the normalized eigenvectors of Π, we are then able to calculate the reduced Laplacian $\overline{L}$ in Equiation (3) and solve the solution Σ of the Lyapunov Equation (2). Finally, the effective resistance between nodes k and j can be calculated through Equation (7.3). The detailed derivation process can be found in [31]. To further extend the definition of effective resistance between some node pairs in any digraph, whether or not it is connected, we have the following definition.

Definition 5. The effective resistance between nodes k and j in a directed graph G is

$$r_{k,j} = \begin{cases} \infty & \text{if there are no connections between nodes } k \text{ and } j \\ r_{k,j} \text{ in } C_G(k,j) & \text{if } C_G(k,j) \text{ exists,} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

We then introduce the edge sampling method, which is based on the effective resistance $r_{k,j}$. Define

$$p_{k,j} = \min(1, C(\log N) r_{k,j}/\epsilon^2)).$$

Here C is some absolute constant, $r_{k,j}$ is the effective resistance between nodes k and j. Then ϵ is the only pruning parameter that determines the number of edges retained in the sparse graph, which in turn corresponds to the quality of approximation after edge sampling. Then the edge sampling algorithm can be conducted simply as if $C_G(k,j)$ exists, we include edge (k,j) in G with the probability $p_{k,j}$.

This edge sampling algorithm runs for every meta-path graph before applying the attention mechanism to them. We also prove that, during this process, the node features learned through attention layers are preserved in each of the meta-path graphs. The detailed proof can be found in Appendix C. Since the edge sampling algorithm is independent of feature aggregations in graph neural networks, we can sample all meta-path subgraphs at first. Then the resistance sampling procedure would not contribute to extra latency time in FastHAND. The whole algorithm of FastHAND is summarized in Algorithm 1.

7.4 Time Complexity Analysis

The computational complexity of our FastHAND method mainly involves two parts. The first one is the effective resistance sampling and the second one is the attention mechanism used in the graph neural networks. Below, we discuss the computational complexity of each part in detail.

Effective Resistance Sampling. To compute the effective resistance between a pair of nodes, denoted as k and j, we employ a bi-directional breadth-first search algorithm to construct their respective connection graph $C_\Phi(k,j)$ within the meta-path graph Φ. This process involves traversing both directions starting from the respective nodes and expanding along all accessible paths until the connection graph is fully established. The computational complexity of this process is $O(|E_\Phi|)$, where $|E_\Phi|$ is the number of meta-paths in Φ. Given that this process can be parallelized, substantial computational time savings can be achieved. Once we have the effective resistances for all node pairs in the meta-path graph, we can then implement the edge sampling algorithm for sparsification, which has a complexity of $O(|V_\Phi^2|)$, where $|V_\Phi|$ is the number of nodes in Φ.

Attention Mechanism. For a meta-path graph Φ and the m-th user feature group s^m, the complexity of subuser-level attention is $O(|V_\Phi||z^m||s^m| + |E_\Phi||z^m|)$, where $|s^m|$ is the dimension of the m-th user feature group and $|z^m|$ is the dimension of subuser-level embedding. Since the sub-user-level and semantic-level attention can be computed in parallel, the overall complexity is linear to the number of users and meta-path-based user pairs.

8 EXPERIMENTS

We emphasize that, the intuition of our study is to investigate the influence of user affinity propagation through regulated social tunnels to social relations. However, the proposed FastHAND method does not explicitly utilize structural insights from social graph. To solve this problem, we formulate it into a link prediction task, since social graph is the major dependency in this task. Specifically, we conduct the following experiments. First, we compare FastHAND with several state-of-the-art methods to demonstrate our intuitions on user taste propagation. We then regard FastHAND as a feature learning layer and use the features learned through FastHAND to replace the original node feature inputs in some competitors to verify their usefulness. Last, we conduct feature ablation study to gain insights about the importance of social behaviors in the NIO network.

Algorithm 1: The overall process of FastHAND Algorithm

Input:

The heterogeneous graph $G = (V, E)$,
The node feature $s_i, i \in V$,
The meta-path graphs $\{\Phi_0, \Phi_1, ..., \Phi_P\}$,
The number of edges to be sampled q,
The number of attention head K,
The number of feature group M,
Labels of social links Y.

Output:

The node embedding,
The MLP weights for link prediction,
The subuser-level attention weight α,
The semantic-level attention weight β.

repeat
  for *each meta-path graph* $\Phi_p \in \{\Phi_0, \Phi_1, ..., \Phi_P\}$ **do**
    for *each attention head* $k \in \{1, 2, ..., K\}$ **do**
      Sample a subgraph $\Phi_p^{'} = EdgeSampleD(\Phi_p, q)$
      for *each feature group* $m \in \{1, 2, ..., M\}$ **do**
        Feature-specific transformation $s'^{m}_{i} = W^{m}_{\Phi_p^{'}} s^{m}_{i}$;
        for $i \in V$ **do**
          Find the meta-path based neighbors $\Gamma_{out, \Phi_p^{'}}(u_i)$;
          for u_j *in* $\Gamma_{out, \Phi_p^{'}}(u_i)$ **do**
            Calculate the weight coefficient $\alpha^{\Phi_p^{'}, m}_{u_i, u_j}$;
          end
          Calculate the subuser-level embedding $z_i^{\Phi_p^{'}, m} = \sigma(\sum_{u_j \in \Gamma_{out, \Phi_p^{'}}(u_i)} \alpha^{\Phi_p^{'}, m}_{u_i, u_j} W^m s^m_j)$;
        end
        Concatenate the learned subuser-level embedding to get user-level attention $z_i^{\Phi_p^{'}} = ||_{m=1}^{M} z_i^{\Phi_p^{'}, m}$;
      end
      Average user-level attention from all attention heads $\tilde{z}_i^{\Phi_p^{'}} = \frac{1}{K} \sum_k z_{i_k}^{\Phi_p^{'}}$;
    end
    Calculate the weight of meta-path $\beta^{\Phi_p^{'}}$;
    Fuse the semantic-specific embedding $Z_i = \sum_{p=1}^{P} \beta^{\Phi_p^{'}} \tilde{z}_i^{\Phi_p^{'}}$;
  end
  Concatenate semantic-specific embedding of node pair (u_i, u_j) to make link predictions $y_{i,j} = \sigma(Z_i || Z_j)$;
  Compute binary cross-entropy loss $\mathcal{L} = -\sum_{(u_i, u_j)} (Y_{(u_i, u_j)} log(y_{(u_i, u_j)}) + (1 - Y_{(u_i, u_j)}) log(1 - y_{(u_i, u_j)}))$;
  Back propagation and update parameters in FastHAND;
until *convergence*;

8.1 Baselines

We compare with some state-of-art baselines designed for predicting directed links, including graph neural network based methods and graph embedding methods, to verify the effectiveness of the proposed FastHAND.

- *Gravity-AE (GAE)* and *Gravity-VAE (GVAE)* [20]. Inspired by asymmetric universal gravity, both models utilize a two-layer GCN encoder with 64-dim hidden layer to encode each node along with its features, into a 32-dim latent vector representation and a mass value. Then the 33-dim embeddings are used to reconstruct original social graph. The two models are trained with Adam optimizer, learning rate 0.2 and early stopping with a patience of 100 epochs.
- *HOPE* [15]. This method embeds each node as a source vector and a target vector. The resulting embeddings are assumed to preserve high-order asymmetric proximity metrics, including Katz Index, Rooted PageRank, Common Neighbor and Adamic-Adar. In this method, we set $\beta = 0.01$, the dimensions of source and target vectors to be 16.
- *APP* [38]. This method utilizes only structure information by explicitly modeling each direct link as the probability $P(v|u)$ for node v given its neighbor u. We train this model over 100 iterations to learn a 32-dim node representation for each node, with standard settings from its original implementation.
- *MLP*. This method is a naive two-layer MLP classifier. The first layer is fully connected and utilizes ReLU activation to map ordered concatenation of user features into a 32-dim hidden vector. The second layer is also fully connected and utilizes Softmax activation to map the hidden vector into a binary identifier.
- *GAE-FastHAND* and *GVAE-FastHAND*. The features acquired through FastHAND are employed to substitute the original inputs of GAE and GVAE, providing additional evidence of FastHAND's effectiveness.

8.2 Implementation Details

We implement the FastHAND algorithm using Python. To compute effective resistance, we leverage the NetworkX library (https://networkx.org/) for graph construction, compute the Laplacian matrix L using its built-in function (networkx.directed_laplacian_matrix), utilize SciPy for solving the Lyapunov Equation (2) (scipy.linalg.solve_continuous_lyapunov), and employ NumPy (https://numpy.org/) for matrix multiplication in Equations (1) and (3). The edge sampling process incorporates two hyper-parameters (C and ϵ) to regulate the number of sampled edges. C is an empirical constant that is set to 0.08, and $\epsilon \in (0, 1]$ is a pruning parameter that determines the quality of approximation after sparsification. Intuitively, a lower value of ϵ would yield denser sparsified graphs and better quality of approximations. Following the approach outlined in [21], we set $\epsilon = 0.5$ for all meta-path graphs, achieving a favorable balance between the quality of approximation and computational speed. For the GNN part, we adhere to common practice in related works by utilizing the Adam SGD optimizer, setting the dropout of attention to 0.6, and early stopping with a patience of 100 epochs. To make a relatively fair comparison with other competitors, we embed each sub-user-level feature group into a 32-dim vector and set the number of attention heads K to 4. In the absence of a consensus on the learning rate, we conduct a sweep in the set $\{0.01.0.005, 0.001, 0.001\}$ and found 0.005 leads to the best performance.

8.3 Three Directed Link Prediction Tasks

Following [20], we evaluate the predictive power of FastHAND in an ordinary task and two challenging tasks. Below, we introduce each task in details.

TASK 1 (General Directed Link Prediction). Following previous works [5, 16, 19, 20, 24], we train models on incomplete versions of graphs, where 15% of edges are randomly removed. To take the direction of links into consideration, when link (i, j) is removed, its reciprocal link (j, i) is not necessarily removed. We then randomly sample some unconnected node pairs as the negative samples, whose number is the same with the removed 15% edges. This constructs a dataset for validation (taking up 2/3) and test (taking up 1/3).

TASK 2 (Biased Negative Samples Link Prediction). In this task, we only remove unidirectional links. Specifically, if a linked pair (i, j) exists in the removed set, then its reciprocal candidates (j, i) is also removed. In fact, those reciprocal ones are unconnected node pairs and we use them to constitute the negative samples. This setting has been proposed in [38] and [20] and is more challenging than *TASK 1*, because it is harder to reconstruct asymmetric social relations.

TASK 3 (Bidirectionality Prediction). This task is formulated following [20] to evaluate the models in discriminating bidirectional links from unidirectional links. We first randomly remove one of the two directions of all bidirectional links, contributing to a training graph with only unidirectional connections. We then evaluate the models in distinguishing between the removed links and node pairs that are unconnected from beginning to end. When more removed links are precisely predicted as positive samples and unconnected pairs labeled as negative ones, the model is stronger in discriminate bidirectional social relations.

8.4 Data Pre-processing

User profile is an important source to derive user features. However, the profiles are mainly generated by users themselves, then the information quality is not high. To solve this problem, we take the following steps to achieve informative and clean experimental data.

Data completion. As we mentioned before, the information of job titles for NIO staff is not completed in the NIO dataset. In addition, the level hierarchy of NIO staffs is unknown. To solve these problems and also utilize the job title information, we develop a novel penalized-pagerank *job rank inference* algorithm; see Appendix A for more details. The NIO enterprise sets up regional branches in different municipals around the mainland China, so that user relations are strongly correlated to geo-labels. However, the information of geo-location is also not complete. To complete geo-locations that are left blank by users, we propose a novel CAMLP label propagation algorithm; see Appendix B for more details.

Data cleaning. From the perspective of constructing user features, we remove users whose headshots and publications are left blank or their profiles are blocked to strangers, since we are not able to fully capture their characteristics. From the perspective of constructing meta-paths, we remove users who appear in less than 2 meta-path graphs, since these users are not active. After these steps, there are $27,556$ users left in the NIO dataset.

8.5 Results of Social Link Prediction

We compare FastHAND with the state-of-the-art methods to demonstrate its predictive power. To evaluate model performance, we use AUC and AP as the metrics. The experimental results in three prediction tasks are listed in Table 4. In *TASK 1*, we find methods utilizing both node features and structural information (such as GAE, GVAE, GAE-FastHAND and GVAE-FastHAND) have achieved better performances than methods only utilizing structural information. Our proposed FastHAND outperforms HOPE only utilizing structural information and MLP only utilizing node features, but less powerful than APP. We also find GAE-FastHAND and GVAE-FastHAND, which utilize propagated user tastes, outperform GAE and GVAE. This result demonstrates the efficiency of our intuition that, regulated social tunnels in EESNs are vital for user interests to propagate. Last, GAE-FastHAND and GVAE-FastHAND also achieve the best performance among all the methods.

In *TASK 2* and *TASK 3*, we find all methods have worse prediction performance than those in *TASK 1*, because these two tasks are more challenging. Among them, the predictive power of gravity-inspired methods (e.g., GAE and GAVE) suffer greater declines than those of their competitors. On the contrary, FastHAND is less sensitive to the change of tasks, because it does not rely on social graph to make predictions. As a result, FastHAND has achieved better prediction performance against the state-of-the-art methods. In addition, we find performance declines of GAE-FastHAND and GVAE-FastHAND are relatively smaller than those of GAE and GVAE, indicating that user features extracted by FastHAND can inherently embed social affinities and thus be effective for friendship predictions. In conclusion, based on these experimental results, we have demonstrated that, the proposed FastHAND method has good ability in modeling user social affinities, behaves effective in solving link prediction tasks, and also shows robustness to task shifts.

Table 4. The comparison of FastHAND with other link prediction methods (i.e., GAE, GVAE, HOPE, APP, and MLP) in three prediction tasks. In addition, we substitute the features from FastHAND for the original ones in GAE and GAVE, which leads to the GAE-FastHAND and GVAE-FastHAND methods. For each method, we compute AUC and AP as the evaluation metrics.

	Metric	GAE	GVAE	HOPE	APP	MLP	FastHAND	GAE-FastHAND	GVAE-FastHAND
Task 1	AUC	0.9747	0.9751	0.7673	0.9031	0.7046	0.8833	0.9778	0.9841
	AP	0.9791	0.9523	0.7315	0.8903	0.6812	0.8716	0.9831	0.9827
Task 2	AUC	0.6172	0.6503	0.5683	0.6748	0.6505	0.7513	0.7237	0.8045
	AP	0.6067	0.6417	0.5619	0.6530	0.6515	0.7354	0.7208	0.7893
Task 3	AUC	0.6422	0.6401	0.6120	0.6983	0.6517	0.7059	0.7797	0.7657
	AP	0.6221	0.6205	0.6322	0.6946	0.6812	0.7132	0.7546	0.7629

8.6 Predictive Power of Features and Meta-Paths

In this section, we make detailed analysis for different features and meta-paths. Specifically, we focus on three feature groups, i.e., the social behavior statistics (SBS), the basic statistics (BS), and the user generated contents (UGC), and seven meta-path graphs. Then we investigate the predictive power of different combinations of feature groups within different meta-path graphs. The detailed results evaluated by AP are shown in Table 5.

To compare the prediction performances of feature group BS, SBS and UGC in each of the meta-path column in Table 5, we find SBS achieves slightly better performances than BS, and both BS and SBS significantly outperform UGC. To take the combinations of feature groups into consideration, we find BS+SBS, BS+UGC, SBS+UGC, and all features (denoted by “All Fts") consistently achieve the best performances in all meta-paths. When BS is combined with SBS, the predictive performance is dramatically increased, when compared with the prediction performance using each single feature group. However, to combine UGC with any other feature groups, we find the marginal effects are small. This finding further indicates that UGC is less influential to social relations.

We then focus on the performance of different meta-path graphs. Note that, different feature groups over each meta-path graph show similar trends. Thus for simplicity, we only compare the results of all features (“All Fts") in different meta-path graphs. Better link prediction performances indicate the meta-paths are good social tunnels and highly informative for users to get acquainted and amiable with each other. It is not surprising that, “All MP" (i.e., all meta-paths) achieves the best performance. Among the rest seven single meta-path graphs, we find the MP_{UatU}, $UA_{off}U$ and MP_{UCCU} achieve the top three performances with the average AP value over 0.82. This result suggests that, meta-paths builting based on stronger social interactions show better performances. For illustration, users who refer to each other in posts or comments, strongly indicate the existences of their offline friendships. Thus the MP_{UatU} meta-path achieves better performance against all single meta-path settings. Besides, users in offline activities spare more time together and are able to meet face to face. Thus the corresponding meta-path $UA_{off}U$ outperforms $UA_{on}U$ in predicting friendship formation. Finally, UGU achieves the worst predictive performance. This result implies that, large user groups including hundreds or thousands of users, are not good platforms for users to build strong ties.

8.7 Feature Ablation Study

In this section, we conduct feature ablation study to investigate influences of some interested user features to social relations. Specifically, we focus on nine features, including car-ownership (BS_{own}), NIO staff or not (BS_{staff}), job rank score ($BS_{jobrank}$), geo-location (BS_{geo}), user headshot ($BS_{headshot}$), link beggar or not (BS_{beggar}), NIO-staff social transcendence (SBS^{staff}_{trans}), car-owner social transcendence (SBS^{own}_{trans}), and user topic distribution (UGC_{topic}).

Table 5. The comparison of predictive power of different combinations of feature groups and meta-paths. We consider three feature groups, i.e., the social behavior statistics (SBS), the basic statistics (BS), and the user generated contents (UGC), along with their combinations. We consider seven meta-paths for each column, the definitions of which can be found in Table 3. “All MPs" represents using all meta-paths. Under each combination of feature group and meta-path, we compute AP as the evaluation metric. The best results are indicated by bold.

	All MPs	$MP_{UA_{on}U}$	$MP_{UA_{off}U}$	MP_{UTU}	MP_{UCPU}	MP_{UCPCU}	MP_{UCCU}	MP_{UatU}
BS	0.7740	0.7291	0.7338	0.7234	0.7212	0.7355	0.7380	0.7501
SBS	0.7949	0.7487	0.7490	0.7251	0.7403	0.7396	0.7432	0.7638
UGC	0.6806	0.6515	0.6315	0.6346	0.6462	0.6323	0.6359	0.6723
BS+SBS	0.8839	0.8022	0.8061	0.7515	0.7942	0.7794	0.8038	0.8432
BS+UGC	0.7801	0.7458	0.7687	0.7602	0.7510	0.7595	0.7497	0.7736
SBS+UGC	0.7953	0.7574	0.7701	0.7985	0.7694	0.7502	0.7561	0.7920
All Fts	**0.8833**	**0.8153**	**0.8356**	**0.8012**	**0.8110**	**0.8071**	**0.8211**	**0.8592**

To explore the prediction power of each feature, we would compare the AP values with or without this feature. Then a downgrade ratio of AP is computed. The higher the ratio, the more important the feature to link prediction. Features with downgrade ratio larger than 0.5% are regarded as influential features. Though predictions are made over the graph among all kinds of users, we separate their social relations into three classes to gain deeper insights. The three classes are: relations among NIO-Staffs, relations among Non-Staffs, and relations between NIO-staffs and Non-Staffs (denoted by InBetween). The detailed downgrade ratio results of all features are shown in Figure 5. Overall, we find BS_{geo} and SBS_{trans}^{staff} are generally more influential to all kinds of social relations than the rest features. Below, we would take a deep analysis for the rest features.

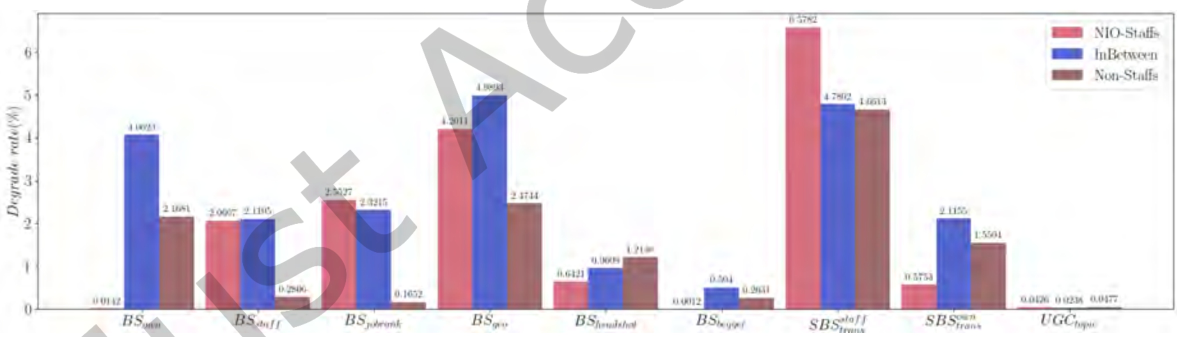

Fig. 5. The degrade rates of AP values when removing specific features.

(1) BS_{own}. The car-ownership tag BS_{own} is less influential to the NIO-Staffs group. The reason behind this finding might be that, NIO-staffs socialize with their colleagues mostly for collaboration purpose, and they do not care much about whether one owns a car or not. Besides, it plays an important role in Non-Staff and InBetween groups. These results demonstrate BS_{own} acts as a useful role in social stratification.

(2) BS_{staff}. This feature is found to be influential for both NIO-Staffs and InBetween groups. This finding is in accordance with our observations that, NIO-staff socialize to corporate and communicate, and they socialize with non-staff users to provide pre-sales consulting and after-sale services. However, knowing both users are non-staff does not help friend decision making.

(3) $BS_{jobrank}$. This feature is a reflection of one's authorities among NIO-staffs. For normal users, their job ranks are set to zero. Similar with BS_{staff}, we find $BS_{jobrank}$ is influential to relations between NIO-Staffs and InBetween groups, but less relevant with Non-Staffs group. The only difference is that, $BS_{jobrank}$ is more influential to NIO-Staffs group than InBetween group. The reason behind this finding is that, when NIO-staffs and normal users decide to follow a NIO-staff, both of them pay attention to his/her authority. In addition, NIO-staffs would follow more strictly along the social hierarchy.

(4) $BS_{headshot}$. This feature is informative in predicting all types of relations, while it exhibits relatively lower influence compared to most other features. This observation may stem from the implicit design of natural image captioning models, which primarily aim to describe factual elements rather than aesthetic attributes like 'young' and 'good-looking'. As a result, they may overlook capturing the subtle nuances of attraction present in headshots, which could potentially represent the true essence of interpersonal attraction.

(5) BS_{beggar}. This feature has almost nothing to do with social relations among the NIO-Staffs group. This is intuitive as there is no "link beggars" observed among NIO-staffs. For normal users, the number of "link beggars" is also small. We only observe 215 normal users who have requested follow-backs. This is why the downgrade ratios of Non-Staffs group and InBetween group are also very small. However, compared with the Non-Staffs group, the downgrade ratio of InBetween is higher, from which we conclude that NIO-Staff users are more inclined to respond to link requests than normal users.

(6) SBS^{own}_{trans}. This feature is less effective than SBS^{staff}_{trans} in predicting each type of links. This phenomenon might due to the fact that, NIO-staffs socialize with each other mostly for work communications, and thus their preferences in car-owner social transcendence are less relevant. Besides, we find SBS^{own}_{trans} is more effective in predicting links in the InBetween group than the Non-Staffs group. This is because NIO-staffs follow car-owners to provide after-sales services and potential-owners to provide pre-sale consulting, which can be successfully captured by this feature.

(7) UGC_{topic}. Although user topics are found vital for social link prediction [2, 17], UGC_{topic} is the least effective feature in our comparisons. This can be attributed to the reason that, NIO forbids post sharing and suppresses timeline views, which prohibits the propagation of UGCs among users. As a result, we are not able to fully capture user interests from their publications.

9 CONCLUSION AND DISCUSSION

In this work, we investigate social behaviors in novel exclusive enterprise social networks (EESNs). By taking the NIO community as the research object, we find novel social tunnels for information propagation. We also formulate the investigation of NIO as a social link prediction problem over meta-path (social tunnel) graphs. To efficiently solve the link prediction problem, we propose the FastHAND algorithm. It firstly applies spectral sparsification over each meta-path graph to compress the scales of computation, then fully utilizes multi-modal user features and heterogeneous social tunnels provided by NIO to make link prediction. Experimental results not only demonstrate the superiority of FastHAND against many state-of-the-art methods, but also verify our observations and intuitions about the uniqueness of EESNs.

To conclude this work, we present some directions for future work. Firstly, in the FastHAND algorithm, all meta-paths currently share the same set of hyper-parameters. Exploring adaptive hyper-parameter selection tailored to different types of meta-paths is worth consideration. Secondly, while we presently assume a single network embedding for each user node, users inherently possess two identities: as followers and as followed individuals. Therefore, incorporating the "source and target embedding" paradigm to effectively capture these dual identities is an important avenue for further exploration. Third, there is potential for substantial enhancement in the processes of feature extraction and meta-path graph construction. The feature ablation study reveals that specific features may have a limited impact on capturing true interpersonal attractions. Additionally, leveraging the abundant multimodal data (images, videos, locations, and identity authentications) within the primary user-generated content

(UGC) of EESNs presents an untapped opportunity. Therefore, optimizing the integration of multimodal data in feature extraction and meta-path construction steps is a promising avenue for further research. Last, to evaluate the performance of FastHAND, we only consider the NIO community for example. More EESN datasets should be further analyzed for validation.

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China (No.72371241, 72171229, 62172014, 62172015, 62272009), fund for building world-class universities (disciplines) of Renmin University of China, Chinese National Statistical Science Research Project (2022LD06), Foundation from Ministry of Education of China (20JZD023), the MOE Project of Key Research Institute of Humanities and Social Sciences (22JJD110001).

REFERENCES

- [1] Mayla Alimam, Emmanuel Bertin, and Noël Crespi. 2015. Enterprise Social Systems: The what, the why, and the how. In *2015 IEEE 17th Conference on Business Informatics*, Vol. 2. IEEE, 9–17.
- [2] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. 2014. Who to follow and why: link prediction with explanations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1266–1275.
- [3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
- [4] Robin IM Dunbar, Valerio Arnaboldi, Marco Conti, and Andrea Passarella. 2015. The structure of online social networks mirrors those in the offline world. *Social networks* 43 (2015), 39–47.
- [5] Aditya Grover, Aaron Zweig, and Stefano Ermon. 2019. Graphite: Iterative generative modeling of graphs. In *International conference on machine learning*. PMLR, 2434–2444.
- [6] Zepeng Huo, Xiao Huang, and Xia Hu. 2018. Link prediction with personalized social influence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
- [7] Yizhu Jiao, Yun Xiong, Jiawei Zhang, and Yangyong Zhu. 2019. Collective link prediction oriented network embedding with hierarchical graph attention. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 419–428.
- [8] Adri Khalis and Amori Yee Mikami. 2018. Talking face-to-Facebook: Associations between online social interactions and offline relationships. *Computers in Human Behavior* 89 (2018), 88–97.
- [9] Thomas N Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *NIPS Workshop on Bayesian Deep Learning* (2016).
- [10] Jiayu Li, Tianyun Zhang, Hao Tian, Shengmin Jin, Makan Fardad, and Reza Zafarani. 2020. Sgcn: A graph sparsifier based on graph convolutional networks. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part I 24*. Springer, 275–287.
- [11] Xin Liu, Mingyu Yan, Lei Deng, Guoqi Li, Xiaochun Ye, Dongrui Fan, Shirui Pan, and Yuan Xie. 2022. Survey on Graph Neural Network Acceleration: An Algorithmic Perspective. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 5521–5529. https://doi.org/10.24963/ijcai.2022/772 Survey Track.
- [12] Yingjie Liu, Shihu Liu, Fusheng Yu, and Xiyang Yang. 2022. Link prediction algorithm based on the initial information contribution of nodes. *Information Sciences* (2022).
- [13] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. Heterogeneous Graph Neural Networks for Malicious Account Detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 2077–2085. https://doi.org/10.1145/3269206.3272010
- [14] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. 2016. A survey of link prediction in complex networks. *ACM computing surveys (CSUR)* 49, 4 (2016), 1–33.
- [15] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 1105–1114.
- [16] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. 2018. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2609–2615.
- [17] Daniele Quercia, Harry Askham, and Jon Crowcroft. 2012. Tweetlda: supervised topic classification and link prediction in twitter. In *Proceedings of the 4th Annual ACM Web Science Conference*. 247–250.
- [18] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Hkx1qkrKPr

[19] Guillaume Salha, Romain Hennequin, Viet Anh Tran, and Michalis Vazirgiannis. 2019. A degeneracy framework for scalable graph autoencoders. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 3353–3359.

[20] Guillaume Salha, Stratis Limnios, Romain Hennequin, Viet-Anh Tran, and Michalis Vazirgiannis. 2019. Gravity-inspired graph autoencoders for directed link prediction. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 589–598.

[21] Rakshith S Srinivasa, Cao Xiao, Lucas Glass, Justin Romberg, and Jimeng Sun. 2020. Fast graph attention networks using effective resistance based graph sparsification. *arXiv preprint arXiv:2006.08796* (2020).

[22] Yizhou Sun and Jiawei Han. 2013. Mining heterogeneous information networks: a structural analysis approach. *Sigkdd Explorations* 14, 2 (2013), 20–28.

[23] Benjamin Wehner, Christian Ritter, and Susanne Leist. 2017. Enterprise social networks: A literature review and research agenda. *Computer Networks* 114 (2017), 125–142.

[24] Max Welling and Thomas N Kipf. 2016. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*.

[25] Lilian Weng, Jacob Ratkiewicz, Nicola Perra, Bruno Gonçalves, Carlos Castillo, Francesco Bonchi, Rossano Schifanella, Filippo Menczer, and Alessandro Flammini. 2013. The role of information diffusion in the evolution of social networks. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 356–364.

[26] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. PMLR, 2048–2057.

[27] Yuto Yamaguchi, Christos Faloutsos, and Hiroyuki Kitagawa. 2016. Camlp: Confidence-aware modulated label propagation. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 513–521.

[28] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web*. 1445–1456.

[29] Yang Ye and Shihao Ji. 2021. Sparse graph attention networks. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2021), 905–916.

[30] Nick Yee, Jeremy N Bailenson, and Nicolas Ducheneaut. 2009. The Proteus effect: Implications of transformed digital self-representation on online and offline behavior. *Communication Research* 36, 2 (2009), 285–312.

[31] George Forrest Young, Luca Scardovi, and Naomi Ehrich Leonard. 2015. A new notion of effective resistance for directed graphs—Part I: Definition and properties. *IEEE Trans. Automat. Control* 61, 7 (2015), 1727–1736.

[32] Seongjun Yun, Seoyoon Kim, Junhyun Lee, Jaewoo Kang, and Hyunwoo J Kim. 2021. Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. *Advances in Neural Information Processing Systems* 34 (2021), 13683–13694.

[33] Jiawei Zhang, Yuanhua Lv, and Philip Yu. 2015. Enterprise Social Link Recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (Melbourne, Australia) *(CIKM '15)*. Association for Computing Machinery, New York, NY, USA, 841–850. https://doi.org/10.1145/2806416.2806549

[34] Jiawei Zhang, Philip S. Yu, and Yuanhua Lv. 2015. Organizational Chart Inference. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia) *(KDD '15)*. Association for Computing Machinery, New York, NY, USA, 1435–1444. https://doi.org/10.1145/2783258.2783266

[35] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in neural information processing systems* 31 (2018).

[36] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. 2021. Data augmentation for graph neural networks. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 35. 11015–11023.

[37] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. 2020. Robust graph representation learning via neural sparsification. In *International Conference on Machine Learning*. PMLR, 11458–11468.

[38] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. 2017. Scalable graph embedding for asymmetric proximity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

APPENDIX A: JOB RANK INFERENCE

In social science, people in the lower level class are eager to get acquainted by the upper level class. The same trend exists in EESNs, as users tend to follow their upper-level managers, rather than subordinates [34]. These observations are in accordance with the underlying assumption of PageRank algorithm that "more important websites are likely to receive more links from other websites[1]". Based on this idea, we apply the PageRank

[1] https://www.google.com/search/howsearchworks/

algorithm to rank authorities of NIO staffs, but with some modifications. First, we conduct a two-step PageRank algorithm. In the first step, we run RageRank for social graph only constructed by NIO staffs and get PR_{NIO} score for every NIO staff. In the second step, we run RageRank for social graph constructed by all users, and then penalize the original PR_{NIO} scores of NIO staffs to get $PR_{penalized}$. Second, there are NIO staffs who have the same job titles but work in different cities or provinces. Then we average scores of these NIO staffs to get rid of area preferences and individual differences. This leads to the finally inferred Job-level ranks for all NIO staffs.

We first evaluate the efficiency of PR_{NIO} scores through an approximation of AUC metric. Specifically, we sample 1,000 pairs of users whose job titles can be directly used to infer their differences of job levels. These pairs of users are generated from the following three scenarios: (1) one from headquarter and one from sub-divisions, (2) the job title explicitly claims its owner to be the superior of others, (3) several top leaders and all their employees. Then we calculate the consistency rate between the manually evaluated job title ranks and the ranks derived from PR_{NIO} scores. As a result, the approximated AUC value is 0.837. For further illustration, we list several job titles and their PageRank scores in Table A.1.

From the "Original Algorithm" column of Table A.1, we find "User Business Service Supervisor" achieves higher PageRank score than the founder, co-founder and VPs of NIO. Besides, the operation management (OM) of regional company achieves higher score than his/her superior general manage (GM). The phenomenon of NIO staffs achieving higher job level ranks than their real statuses is due to the fact that, these NIO staffs behave more like normal users and thus gain more attentions from their colleagues. To solve this problem, we utilize the PageRank scores of NIO staffs in the entire community to penalize this user affinity phenomenon. That is, when a NIO staff gains higher attentions from the general public, he/she will suffer bigger penalties. Specifically, the penalized PageRank score is computed as follows.

$$PR_{penalized} = (PR_{NIO} - c_1 * PR_{All})/(1 + e^{1/c_2}),$$

where c_1 is the experience penalty coefficient between 0 and 1. The original job level rank c_2 is also utilized to set decay factor $1/(1 + \exp 1/c_2)$ to penalize higher ranked NIO staffs. Intuitively, higher the job level rank, larger the decay factor. As a result, the AUC value of penalized PageRank algorithm grows from 0.837 to 0.901. The improvement can also be validated in column "Penalized Algorithm" of Table A.1, where the rank of "Founder, Chairman, CEO" achieves the first place and "Regional GM" surpasses "Regional OM". Finally, we use the penalised PageRank scores for NIO staffs as the job level feature; and for normal users, we set their job level scores to be 0.

Table A.1. The top six ranked job titles in original and penalized PageRank algorithms. All values are $\times 10^6$

Job Title	Original Algorithm		Penalized Algorithm	
	PR_{NIO}	Rank(c_2)	$PR_{penalized}$	Rank
User biz Service Supv	9479.33	1	96.64	2
Founder, Chairman, CEO	6492.55	2	123.47	1
Co-founder, President	3581.41	3	79.48	3
Vice President	110.25	4	4.90	4
Regional OM	599.94	5	2.99	6
Regional GM	482.44	6	4.18	5

APPENDIX B: GEO-LOCATION INFERENCE

Social activities of users in NIO are strongly correlated with their geographical locations. The NIO enterprise builds independent organizations in different administrative areas to provide differentiated services with local characteristics. To verify this idea, we draw the heapmap of averaged social links per user from 31 cities and provinces in mainland China. As shown by Figure B.1, the vertical axis and horizontal axis list each province, the name of which are abbreviated by capital letters (e.g., GD refers to Guangdong providence). The order of the 31 cities and provinces are consistent with the descending order of their 2021 annual GDPs. Colors over the map show the averaged number of social link per user, and the higher the value, the deeper the color.

From the heatmap, we find several interesting phenomenons. First, the heatmap is almost symmetrical along the diagonal line, which means users generally send out and receive the same amount of followships in all areas. Second, colors in the top and left areas are generally deeper than in other areas, which means users in well developed areas are more active in making friends with others. Third, colors in the diagonal line are generally deeper than in other areas, which means users prefer to follow local users rather than those in other areas. Last, the color of entries associate with Shanghai (SH) and Beijing (BJ) are deeper than others, which means users in these two cities are more active than others. In summary, geo-locations of pairs of users serve as strong indicator for the formation of social links.

However, there are 18.38% of geo-locations in user profiles left blank. To complete this information, we use the CAMLP label propagation algorithm to inference geo-locations of the unobserved ones. Instead of directly applying CAMLP to the NIO dataset, we revise this algorithm from three perspectives. First, the "confidence-aware" mechanism of CAMLP permits any label to change when a node has higher confidence of belonging to another label. However, we fix part of the labels, i.e., the observed geo-labels of NIO staffs, since the enterprise staffs are more careful when filling up their profiles. Second, we calculate a 31×31 preference matrix to denote regional preferences among 31 areas, rather than directly using a default diagonal matrix. Finally, for "signal" $s_{ij}(k)$, which means how strongly node i believes node j to have label k, we triple its value when node i is a NIO staff belonging to the last three job levels. This is because lower level staffs usually have more face to face contact with other users in offline scenarios and their geo-labels are more trustworthy. As a result, the modified CAMLP algorithm significantly outperforms the original one. When predicting labels of NIO staffs, it achieves 83.91% accuracy, 12.01% higher than that of the original CAMLP algorithm. In terms of predicting the general customers, it achieves 75.53% accuracy, 10.97% higher than that of the original CAMLP algorithm.

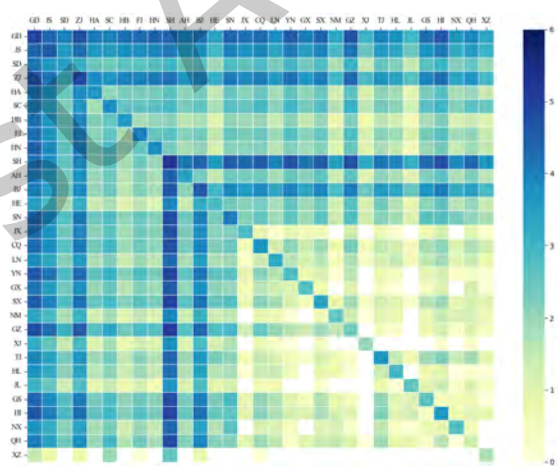

Fig. B.1. The heatmap of averaged number of social links per user in 31 municipalities and provinces in mainland China. All values are in logarithm transformation.

APPENDIX C: PROOF FOR RESISTANCE EDGE SAMPLING METHOD

We aim to prove that, the features learned by FastHAND can be preserved when using the resistance edge sampling method. Let $G^{\sigma} = (V, W)$ be a directed and weighted graph, where $V = \{v_1, ..., v_N\}$ is a set of nodes or vertices and $W = \{w_{ij} : 1 \leq i, j \leq N\}$ is a weight matrix. Usually, we have $w_{ij} \geq 0$ for all $i, j \in \{1, ..., N\}$, and $w_{ii} = 0$ for $i = 1, ..., N$. A set $\{v_i, v_j\}$ is an edge if $w_{ij} \geq 0$. The corresponding (undirected) graph $G = (V, E)$ with $E = \{v_i, v_j | w_{ij} \geq 0\}$ is called the underlying graph of G^{σ}.

Given the directed and weighted graph $G^{\sigma} = (V, W)$, if $\{e_1, ..., e_m\}$ are the total of m edges of the underlying graph G, then the incidence matrix B^{σ} of G^{σ} is an $N \times m$ matrix whose entries b_{ij} are given by

$$b_{ij} = \begin{cases} +\sqrt{w_{ij}} & if\, s(e_j) = v_i \\ -\sqrt{w_{ij}} & if\, t(e_j) = v_i \\ 0 & otherwise. \end{cases}$$

Further denote D to be the degree matrix associated with W, whose diagonal elements are the degree of each node and the non-diagonal elements are all zero. Then we have $B^{\sigma}(B^{\sigma})^{\top} = D - W = L$. Consequently, $B^{\sigma}(B^{\sigma})^{\top} = D - W = L$ is independent of the orientation of G and $L = D - W$ is symmetric and positive semi-definite. That is, the eigenvalues of L are real and non-negative.

For a directed and weighted graph $G^{\sigma} = (V, W)$ without isolated vertex, define its (normalized) graph Laplacians L_{norm} and L_{rw} as follows

$$L_{norm} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2},$$
$$L_{rw} = D^{-1} L = I - D^{-1} W = D^{-1/2} L_{norm} D^{1/2}.$$

Then it is easy to derive that, for any $x \in \mathbb{R}^m$ with $x \neq \mathbf{0}$, we have

$$x^T L_{norm} x = \frac{1}{2} \sum_{i,j=1}^{m} w_{ij} (\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})^2.$$

Where d_i corresponds to ith diagonal element in D. Consequently, $x^T L_{norm} x$ does not depend on diagonal entries in W, and $w_{i,j} \geq 0$ for all $i, j \in 1, ..., m$, then the matrix L_{norm} is positive semi-definite. Besides, since L and $D^{-1/2}$ are symmetric, the L_{norm} is also a symmetric matrix. Similarly, we also have L_{rw} to be symmetric and positive semi-definite.

Let $\|L_{rw}\|$ be the normalized random walk matrix of the original graph and $\left\|L_{rw,s}\right\|$ be the normalized random walk Laplacian matrix of the spectrally sparsified graph through resistance sampling. Then based on the conclusion that both L_{norm} and L_{rw} are symmetric, positive and semi-definite matrices, and also according to [21], we know the graph attention network can be approximated by sparser graphs, i.e.,

$$\left\|L_{rw} - L_{rw,s}\right\| \leq 6\epsilon \left\|L_{norm}\right\|,$$

where ϵ is a constant between 0 and 1. Consequently, the features learned by FastHAND can be preserved when using the resistance edge sampling method.