Evolving Influence Maximization in Evolving Networks

XUDONG WU, LUOYI FU, ZIXIN ZHANG, HUAN LONG, JINGFAN MENG, XINBING WANG, and GUIHAI CHEN, Shanghai Jiao Tong University, China

Influence Maximization (IM) aims to maximize the number of people that become aware of a product by finding the “best” set of “seed” users to initiate the product advertisement. Unlike most prior arts on the static networks containing fixed number of users, we study the evolving IM in more realistic evolving networks with temporally growing topology. The task of evolving IM, however, is far more challenging over static cases in the sense that the seed selection should consider its impact on future users who will join network during influence diffusion and the probabilities that users influence one another also evolve over time.

We address the challenges brought by network evolution through EIM, a newly proposed bandit-based framework that alternates between seed nodes selection and knowledge (i.e., nodes’ growing speed and evolving activation probabilities) learning during network evolution. Remarkably, the EIM framework involves three novel components to handle the uncertainties brought by evolution: (1) A fully adaptive particle learning of nodes’ growing speed for accurately estimating future influenced size, with real growing behaviors delineated by a set of weighted particles. (2) A bandit-based refining method with growing arms to cope with the evolving activation probabilities via growing edges from previous influence diffusion feedbacks. (3) Evo-IMM, an evolving seed selection algorithm, which leverages the Influence Maximization via Martingale (IMM) framework, with the objective to maximize the influence spread to highly attractive users during evolution. Theoretically, the EIM framework returns a regret bound that provably maintains its sublinearity with respect to the growing network size. Empirically, the effectiveness of the EIM framework is also validated with three notable million-scale evolving network datasets possessing complete social relationships and nodes’ joining time. The results confirm the superiority of the EIM framework in terms of an up to 50% larger influenced size over four static baselines.

CCS Concepts: • Computer systems organization → Embedded systems; Redundancy; Robotics; • Networks → Network reliability;

Additional Key Words and Phrases: Evolving social network, evolving influence maximization, multi-arm bandit
1 INTRODUCTION

With the development of massive social networks (e.g., Facebook, Wechat, and Twitter), Influence Maximization (IM) has become a key technology of viral marketing in modern business [1–8]. Given a social network $G$ and an integer $K$, the goal of IM is to select $K$ seed users in $G$ in hope that their adoptions of a promoted product or idea can maximize the expected number of final adopted users through word-of-mouth effect [9, 15]. Initially put forwarded by Kempe et al. [12], the problem of IM has been intensively studied by a plethora of subsequent works, proposing improvements or modifications from multiple aspects, including influence diffusion size estimation [8, 13], adaptive seeding [6, 10], boosting seeding [3], and many others.

The common solutions for the IM problem model the given social network as a graph, where the nodes represent users and the edges represent the social links among users. Each edge is associated with an activation probability referring to the probability that a user can successfully influence his social neighbor after having been influenced himself. The fundamental task in IM lies in estimating the expected influenced size of each alternative seed set based on the network topology and activation probabilities. While existing literatures work well in finding the most influential seed users, they are mainly constrained to the assumption that the number of nodes in the network, along with their edges in between, are fixed during influence diffusion. Consequently, it violates real practices, as many realistic social networks are usually growing over time. Take Wechat [16], the most popular social media app in China, as an example. The number of Wechat accounts (nodes) grew from zero to 300M during its early two years, with 410K new users per day on average, and are continuing to quickly approaching almost 1B users [17]. Moreover, Facebook also exhibits a fast growth with roughly 340K new users per day [18]. Similar phenomena also hold in a wide range of other real social applications including Twitter, academic networks, and so on. Meanwhile, a viral marketing action such as the web advertisements via messages or emails propagation may consume up weeks to months [19]. Thus, given an evolving network $G_t$ at time $t$ and time span $T$ for a viral marketing action, $G_t$ has greatly evolved to $G_{t+T}$ during influence diffused from seed users to the expected maximal size. Consequently, the expected influenced size estimated by existing IM techniques over $G_t$ cannot reflect the influence of seed set over $G_{t+T}$, which severely impacts the quality of selected seed users.

The above issue motivates the study of evolving influence maximization (EIM), whose problem formulation should incorporate the evolutionary nature of $G$ during diffusion. Interpreted technically, given an instance of evolving social network $G_t$ at time $t$ and an integer $K$, the goal of the EIM problem is to select $K$ seed users to maximize the influence diffused to both existing users and those who will join during time $t$ to $t + T$. Different from the well-investigated existing IM problems, the task of the EIM problem turns out to be highly non-trivial due to the following three challenges in reality: (1) The growing speed of a specific network exhibits uncertainties due to multiple external factors (e.g., the number of potential users, user interests, and peer competitions). Such uncertain growing speed hinders accurately predicting how the network topology evolves during time from $t$ to $t + T$. (2) There is no prior knowledge about the activation probabilities via newly emerged edges, and they may also evolve over time with the changes of social relations among users (e.g., from friends to strangers or on the contrast). Although some recent efforts
have been dedicated to the online IM or multi-round IM where activation probabilities among users are uncertain, the underlying network topology is still assumed to be completely known, thus inapplicable to the evolving situations with both growing nodes and edges. (3) In evolving networks, newly added users are more inclined to establish relationship with those of higher popularity. Thus, users in $G_t$ have different attractiveness to new users, as opposed to existing IM studies that treat each user equally. With the above challenges, it remains open how to effectively solve the EIM problem that can jointly deal with the unknown network topology, uncertain activation probabilities, and heterogeneous attractiveness.

This motivates us to study the EIM problem in this article. By proving its NP-hardness, we attempt to solve the above three challenges in the EIM problem by EIM, a new and novel bandit-based Evolving Influence Maximization framework with multiple rounds of IM campaigns. We adopt the multiple rounds in this article for the aim of learning the knowledge about the network’s growing speed and the evolving activation probabilities during the IM campaigns and maximizing the influence diffused to the future users in a long run. Concretely, each round amounts to an IM campaign that chooses seeds that lead to a large diffusion to both existing users and those that will join till the end of this round and improve the network knowledge through interacting with the evolving network. In addition, due to the lack of the full network knowledge, each round incurs a regret on influenced size. Thus, the EIM framework seeks to minimize the accumulated regret incurred by choosing suboptimal seeds over multiple rounds.

While we defer the details of the EIM framework in later sections, here, we briefly unfold its three novel components in addressing the aforementioned three challenges in the EIM problem:

1. It is unrealistic to assume the complete future network topology is known in advance, thus a fully adaptive particle learning-based method is proposed to capture the uncertain network growing speed, with real growing function of node size explicitly represented by a set of weighted particles. By modeling network evolution via the popular Preferential Attachment (PA) rule (i.e., new users preferring connecting to higher degree nodes), we are able to predict the evolving network topology during influence diffusion with weighted particles (Section 4).

2. For coping with the evolving activation probabilities, we model the continuously emerging edges as the growing arms in the bandits, thus ensuring the applicability of the EIM framework to the evolving network with growing nodes and edges (Section 5). By modeling the activating probabilities as the reward distributions of the arms, the reward of each arm as the edge-level feedback can then be taken to adaptively refine the estimated values of evolving activation probabilities.

3. Aiming at maximizing the influence diffused to both existing and future users, we introduce a novel evolving seed selection algorithm Evo-IMM, which leverages the influence maximization via martingale (IMM) framework. The core of the Evo-IMM lies on attaching weights to the nodes such that the weights incorporate the heterogeneity of users’ attractiveness to future users formed by the PA rule (Section 6). In Evo-IMM, users with higher weights are sampled with higher priority in the sampling-based influenced size estimation method. Evo-IMM turns out to enjoy comparable approximation ratio and time complexity with the static counterparts.

We validate the performance of the EIM framework from both theoretical and empirical perspectives. Theoretically, although the growing size and successive emerging orders of the arms due to network evolution further challenge the knowledge learning compared to classical bandits, the regret bound of the EIM framework still provably maintains to be sublinear to the number of periods under the growing network size (Section 7). Empirically, the effectiveness of the EIM framework is validated on both synthetic and real-world evolving networks, with up to 200 years of time span and million scale data size, respectively (Section 8). Notably, the real evolving networks are extracted from the true academic networks with complete co-authorship, citation, and joining time of all authors and papers, which is severely lacking in existing IM works. Experimental results
demonstrate the superiority of the EIM framework. For example, the EIM framework achieves a 50% larger influenced size than four static baselines in an evolving Co-author network with 1.7M nodes.

2 BACKGROUND AND RELATED WORKS

2.1 Static Influence Maximization

Kempe et al. [12] are the first to formulate the influence maximization problem over a given network as a combinatorial optimization problem. Particularly, in their seminal work [12], they treat the network as a graph $G = (V, E)$, where there is an influence cascading process triggered by a small number of influenced users that are called seed users. The influence diffusion process is then characterized by the later widely adopted Independent Cascading (IC) model [3–12], whose definition is given as follows:

Definition 2.1 (Independent Cascading (IC) model). In the IC model, the influence diffusion among users is characterized by the activation probabilities via social links in steps. The selected seed users are initially influenced at step 0. Then, in the following steps, once user $u_i$ is influenced, he remains influenced until the end and has a single chance to activate his each uninfluenced social neighbor $u_j$ successfully with activation probability $w_{ij}$ via the edge (social link) between them in the next step. Such diffusion process stops when no new user gets influenced. Whether or not $u_i$ can influence $u_j$ successfully is independent of the history of the diffusion process. Then, for a given seed set $S$, $I(S, G)$ denotes the expected number of users that are finally influenced by the seed users in set $S$ under the IC model.

With the above IC model, the objective of IM is to find a set of $K$ seed users (i.e., $S^{opt}$) who can maximize $I(S, G)$ among all the sets of users with size $K$. That is,

$$S^{opt} = \arg \max_{S \subseteq V, |S|=K} I(S, G).$$

(1)

Based on the above formulation, Kempe et al. prove the NP-hardness of the IM problem and design the greedy algorithm that provably returns a $(1 - 1/e - \epsilon)$-approximate solution for seed selection. Since then, a large number of subsequent works have emerged to improve the efficiency and quality of seed selection. For some representative examples, References [8, 14, 15] focus on achieving reasonable complexity in seed selection over million- or even billion-scale networks. Besides, different costs for seeding different users are considered in References [6, 23] for the cost-aware IM problem.

The objectives of the above works are mainly set to select the seed set with the maximum $I(S, G)$ estimated over a static network $G$. As a result, over evolving networks where new users continuously join in and activation probabilities evolve over time, it is difficult for classical IM techniques to return high-quality seeds, since $I(S, G)$ estimated by them fails to include the influence diffusion among future users.

2.2 Dynamic and Multi-round Influence Maximization

2.2.1 Dynamic IM. As a step ahead of classical IM problems, some recent attempts are made in dynamic networks. For example, considering the network with changing edges, Reference [20] takes multiple specific examples to show the effect of changing topologies on IM design and highlights the importance of when to activate the seed users. Similarly, the effect of dynamic user availability is studied in Reference [24], and the effect of the activating time is also experimentally shown. Besides, Michalski et al. [25] focus on maximizing the influence diffused to multiple
given network snapshots. However, the future network is assumed to be known in advance, which violates the real practices.

### 2.2.2 Multi-round IM

Meanwhile, there emerges a class of multi-round or online IM techniques that periodically seed one or more users, in a similar manner to our settings that will be described later. To unfold, Tong et al. [5] propose to successively select seed users along with the influence diffusion over dynamic networks while just considering the changing edges among fixed users. In addition, considering the Multi-Arm Bandits (MAB) is a widely used framework that learns dynamics and makes reasonable decision as possible [2], the bandit-based learning framework is adopted in References [2, 10, 21] to learn unknown activation probabilities from the feedbacks of previous influence diffusions and repeatedly select seed users based on the learned activation probabilities. In addition, Sun et al. [26] study the novel problem of adaptively selecting seeds in each round for maximizing the size of users that are influenced in at least one round, and Lin et al. [27] propose to adaptively select seeds in each round with the aim of maximizing the diffusion size of the desired opinion when there are multiple competitive opinions simultaneously diffusing over the network. Regardless of their progress, such online IM works consider the static network topology, where the estimated \( I(S, G) \) also fails to include the influence diffused to future users. Thus, it is still difficult for the seeds to be repeatedly selected at different time to meet the requirement of high quality.

As far as we know, the only work that shares the closest correlation with ours belongs to Li et al. [18], who simulate the network growth based on the Forest Fire Model and then run existing static IM algorithms over the simulated network. However, under the unknown network growing speed, it is difficult for the simulation to capture the real network evolution. Furthermore, the activation probabilities among users are still preset as known constants.

### 2.3 Related Works on Learning Network Evolution and Activation Probability

For coping with the unknown network topology and uncertain activation probabilities in IM, our work is also related to the literatures on learning network evolution and activation probabilities. On one hand, the famous Preferential Attachment (PA) rule, which regulates how new nodes establish social links with existing nodes, is given in References [28, 29] for reproducing the statistical properties of network topologies such as the power-law degree distribution. However, for analyzing the influence diffused to future users, we further need to estimate how many new users will join network during diffusion. Recently, Zang et al. study the patterns of growing network size [17] and individual's connectivity during evolution [30] and give a general function that fits the patterns of the growing size of real social networks. Although the accuracy of the general function is experimentally justified with many real social network datasets, the parameters in the function for a specific network remain unknown in advance.

On the other hand, Leskovec et al. [31, 32] and Shen et al. [33] provide the approaches for inferring the underlying static network topology from the observed activation time of users during diffusion. That is, they focus on inferring the existences of the social links among users. Differently, our work aims at estimating the activation probabilities via social links. Furthermore, Lamprier et al. [34] infer the activation probabilities via social links in a static network through maximizing the likelihood of the action logs of previous diffusion processes. However, in evolving networks, there are no action logs of the diffusion via newly established social links. Moreover, recent works on multi-round IM [2, 10, 21] propose to iteratively refine the estimated values of activation probabilities in each round based on the observed activation feedbacks during previous rounds. Nevertheless, such multi-round IM works still rely on the assumption of fixed topology and activation probabilities.

ACM Transactions on Internet Technology, Vol. 20, No. 4, Article 40. Publication date: October 2020.
The limitations of the related literatures motivate us to study evolving IM, which will be formally defined in next section.

3 EVOLVING INFLUENCE MAXIMIZATION

3.1 Evolving Network Model
We assume that time is divided into different timestamps. Then an evolving network at timestamp $t$ is modeled as a graph $G_t = (V_t, E_t)$, where $V_t$ and $E_t$, respectively, denote the users existing in the network at timestamp $t$ and their social links in $G_t$. We use $d_n^t$ to denote the degree of node $u_n \in V_t$ at timestamp $t$. Given an IM campaign that takes $T$ time (which is called as survival time later), the network may evolve from $G_t$ to $G_{T+t}$ during influence diffusion with newly added nodes and edges. Furthermore, we adopt the well-known Barabási-Albert (BA) model [28, 29] to characterize the evolution process of social networks. BA model is capable of well capturing the typical features such as power-law degree distribution that exist in most real social networks.

Network evolution process. The evolution under BA model is interpreted as follows: a new node joins in the network during each evolving slot $\Delta t$ and establishes $m$ new edges with the existing nodes ($m$ is a constant) [17, 29]. For a newly added user at time $t$, it establishes its first new edge with a chosen existing user $u_n$ according to the rule of Preferential Attachment (PA), which specifies that the probability of choosing $u_n$ is proportional to its current degree, i.e., the value of $\frac{d_n^t}{\sum_{u_j \in V_t} d_j^t + 1}$. By this, given $d_n^t$ being the degree of $u_n \in V_t$ at timestamp $t$, the expected degree of $u_n$ becomes $d_n^t \cdot (1 + \frac{1}{\sum_{u_j \in V_t} d_j^t + 1})$. Then, the remaining $(m - 1)$ edges of the new node are sequentially established in the same manner. Thus, the expected degree of $u_n$ at timestamp $t + \Delta t$ is given by

$$\mathbb{E}(d_{n}^{t+\Delta t}) = d_n^t \cdot \prod_{s=1}^{m} \left(1 + \frac{1}{\sum_{u_j \in V_t} d_j^t + (2s - 1)}\right).$$

Relation between evolving slot $\Delta t$ and the IM campaign survival time $T$. In addition, let $n(t)$ denote the growing size of users over time, then there are $[n(T + t) - n(t)]$ newly joined nodes from time $t$ to $t + T$. Thus, for an IM campaign starting from time $t$, we have the quantitative relation that $T = [n(t + T) - n(t)] \cdot \Delta t$. That is, if there are $[n(t + T) - n(t)]$ new users joining the network during an IM campaign starting from $t$, then there are $[n(t + T) - n(t)]$ evolving slots during the time from $t$ to $t + T$.

Remark. Although $m$ is set as a constant in the BA model [28], it can still well capture the evolution of most networks, with the statistical property of real social networks being that each newly added node expectedly establishes a same number of new edges [17]. The BA model will also be empirically justified in Section 8 under various real datasets, all of which exhibit the phenomenon of “Richer gets richer.”

3.2 Evolving Influence Diffusion Model
Similar to the common settings in existing online IM works, we consider the multiple rounds of IM campaigns. The time of each round corresponds to the survival time $T$ of an IM campaign, and there are totally $R$ rounds. We will further elaborate the reason for the usage of multiple rounds in Section 3.4. Among the $R$ rounds, we let the $r$th round refer to the time from $T^r$ to $T^{r+1}$ (i.e., $[T^r, T^{r+1}]$). The network evolves from $G^r = (V^r, E^r)$ to $G^{r+1} = (V^{r+1}, E^{r+1})$ during $[T^r, T^{r+1}]$. During each round, we characterize the influence diffusion process starting from the selected seed users referring to the IC model in Definition 2.1. However, for relating the two processes of network evolution and influence diffusion, we extend the IC model into the evolving network as follows:
Evolving influence diffusion model. In the $r$th round, we let $w_{ij,r}$ denote the activation probability via the edge from $u_i$ to $u_j$ during time $[T^r, T^{r+1})$. Then, for a node $u_i \in V^{r+1}$, we classify his neighbor nodes until time $T^{r+1}$ into two classifications. The first classification includes the neighbors that establish social links with $u_i$ before $u_i$ getting influenced, and the second classification consists of the neighbors that connect with $u_i$ not earlier than $u_i$ getting influenced. Referring to the IC model, once $u_i$ gets influenced, he then has a single chance to activate his uninfluenced neighbors belonging to the first classification. For each neighbor $u_j$ in the second classification, $u_i$ will have a single chance to activate $u_j$ after $u_j$ connecting with $u_i$ successfully with the probability $w_{ij,r}$. Our insight for such definition is that a new follower of $u_i$ in Twitter may skim through the previous tweets of $u_i$. That is, if a node gets influenced during the $r$th round, he then has the chance to activate each of his uninfluenced neighbors joining the network before $T^{r+1}$. By this, the above evolving influence diffusion model can also be taken as applying the IC diffusion model (Definition 2.1) on the network $G^{r+1}$ at the end of the $r$th round.

In the traditional settings of the IC model, $w_{ij,r}$ remains known and constant during all the $R$ rounds, as it can be determined from the action logs of previous diffusion processes in social medias. However, such traditional IC model cannot be directly applied to the evolving network, since: (1) The activation probabilities via newly established edges remain unknown in advance; (2) The activation probabilities may exhibit random dynamics with network evolution. The reason behind is that real-world factors such as the closeness of user relations and the interests of users in an information with specific contents may be dynamic in evolution [10]. Jointly considering such features and the multi-round setting, in this article, we discretize the variations of the activation probabilities and characterize each activation probability $w_{ij,r}$ as a Gaussian random walk presented below.

Evolving activation probabilities. Let $w_{ij,r}$ denote the activation probability from $u_i$ to $u_j$ during the $r$th round, the value of $w_{ij,r}$ during the $R$ rounds is defined with a Markov process, i.e.,

$$w_{ij,r} = w_{ij,r-1} + v_{ij,r}, \quad v_{ij,r} \sim \mathcal{N}(0, \Delta \Sigma_{ij,r}). \tag{3}$$

Here, $v_{ij,r}$ denotes the Gaussian random noise that characterizes the variation of $w_{ij,r}$ in the $r$th round, and $\Delta \Sigma_{ij,r} = \frac{\Sigma_0}{(r-r_{ij,0})^k} (k > 0)$ specifies the decaying variation. We use $r_{ij,0}$ to record during which round the edge establishes. Specifically, for the new edges that establish during the $(r - 1)$-th round, we initially set that $w_{ij,r-1} \sim \mathcal{N}(\overline{w_0}, \Sigma_0)$ and $r_{ij,0} = r - 1$.

Corresponding to the reality, new edges are established when users make new friends, and the activation probabilities between them may increase over time with them becoming stable close friends. In contrast, a pair of partners may drift apart after their cooperation ends. Besides, the interests of users in an information with specific contents may also increase or decrease over time. Resulting from the both dynamic social relations and interests, the activation probabilities in evolution may randomly become larger or lower over time with decaying variations. With such insight, we adopt the Gaussian noise $v_{ij,r}$ in Equation (3) to quantify the evolution between the activation probabilities in two successive rounds.

3.3 Problem Formulation

With above evolving network and influence diffusion models, the objective of the Evolving Influence Maximization (EIM) problem is selecting a set of seed users at the beginning of each round to maximize the influence diffusion size over the $R$ rounds. We let $S_r$ denote a seed set in the $r$th round, and let $I(S_r, G^{r+1})$ denote the expected size of users that are finally influenced by $S_r$ until $T^{r+1}$. Then, we formally give the EIM problem as below.
Definition 3.1 (EIM Problem). Given the evolving network $G^r$ at the beginning of the $r$th round, where $1 \leq r \leq R$, the objective of the EIM problem in the $r$th round is to find a set of users $S_r^{opt}$ with size $K$ to maximize the expected influence diffusion size $I(S_r, G^{r+1})$. That is, we aim at solving

$$S_r^{opt} = \arg\max_{S_r \subseteq V^r, |S_r| = K} I(S_r, G^{r+1}) \quad (1 \leq r \leq R).$$

Note that in Definition 3.1, the seeds in each round are selected from the current network $G^r$ instead of the future instances $G^{r'}$ ($T^r < t' < T^{r+1}$). The reason behind is that the current network $G^r$ is known, while it is difficult to know which users will be in the future network instances and how they will connect with each other. Since assuming the future instances is known in advance is unrealistic, it is more reasonable to select the seed set from the current $G^r$. Similar to the classical IM problem, the EIM problem in each round is also NP-hard. Lemma 3.2 states the hardness of the EIM problem and the properties of its objective function $I(S_r, G^{r+1})$.

Lemma 3.2. The EIM problem is NP-hard. The computation of $I(S_r, G^{r+1})$ is $\#P$-hard. Then the objective function $I(S_r, G^{r+1})$ is monotone and submodular.\(^1\)

Proof. The NP-hardness and $\#P$-hardness can be, respectively, proved by the reductions from the NP-complete Set Cover problem and the $\#P$-complete S-D connectivities counting problem. The submodularity of $I(S_r, G^{r+1})$ can be proved by modeling the additional influenced size brought by a new seed as the marginal gain from adding an element to the set $S_r$. We leave the detailed analysis in the Appendix A in the supplementary material [42].

Challenges of solving EIM problem. The NP-hardness of EIM implies the necessity to seek for approximate algorithms for seed selection. The typical solution for the classical IM problem in Equation (1) is first estimating the influenced size $I(S, G)$ via simulating or sampling the influence diffusion process under IC model over $G$ and then selecting seeds for maximizing the estimated influenced size in a greedy manner. However, as noted in Section 1, solving the EIM problem is far more challenging due to the evolving nature of the network included. Under Definition 3.1, the three challenges can be reproduced as: (1) At the beginning of the $r$th round, the topology of the network $G^{r+1}$ is unknown. Although the adopted BA evolving model can characterize the potential evolution process, the number of new nodes who will join network from time $T^r$ to $T^{r+1}$ is still required for making predictions on the topology of $G^{r+1}$. Whereas, the node size growing speed $n(t)$ for a given network remains unknown in advance. (2) The activation probabilities among users evolve over time, which, together with the unknown growing speed, renders it impossible to accurately estimate $I(S_r, G^{r+1})$. (3) The heterogeneous attractiveness of users to future users infers that users in $V^r$ cannot be equally treated in diffusion size estimation and seed selection.

3.4 Overview of the Proposed Framework EIM

Regarding the three challenges above, we propose a new framework EIM that can better incorporate the evolving nature in solving the EIM problem.

3.4.1 Basic Idea of Solving EIM Problem. Our solution is built upon the multiple rounds in the EIM problem. The reason for considering the multi-round in the EIM problem is two-fold. On one hand, the survival time of an IM campaign only varies from weeks to months in reality, leading to users joining in the network several months later unable to be influenced by this early IM campaign. Consequently, only selecting the seed users in the beginning and triggering an IM

\(^1\)A set function $I(\cdot)$ is monotone if $I(A) \leq I(B)$ for all $A \subseteq B$, and $I(\cdot)$ is submodular if $I(A \cup x) - I(A) \geq I(B \cup x) - I(B)$ for all $x \notin A \cup B$.  

ACM Transactions on Internet Technology, Vol. 20, No. 4, Article 40. Publication date: October 2020.
Fig. 1. Overview of the EIM framework in the rth trial.

campaign once under the uncertain network knowledge will severely restrict the long-term profits obtained from viral marketing. On the other hand, the periodical seed selection also enables us to cope with the three challenges. To elaborate, the users joining during previous rounds are the natural samples to learn the growing speed. Furthermore, the evolving activation probabilities among users can be learned from the activating results during the influence diffusion in previous rounds. Therefore, to systematically solve the above three challenges, our solution consists of the following three steps in each round: (1) Learning network growing speed from the feedbacks of observed newly added users and then predicting the topology of network until the end of the round. (2) Learning evolving activation probabilities from previous influence diffusion feedbacks. (3) Selecting seed set for triggering an IM campaign under the refined network knowledge in above two steps. Taking the rth round as an example, the objective of step (1) is to predict the network topology until $T_{r+1}$, and step (2) aims at learning the activation probabilities among users to accurately estimate $I(S_r, G_{r+1})$. Then step (3) focuses on selecting a set $S_r$ of seeds who can maximize $I(S_r, G_{r+1})$.

While we unfold the details of the three steps in Sections 4, 5, and 6, respectively, we remark here that the idea of periodical seed selection for the EIM problem cannot be trivially extended from that in recent online IM studies. As pointed out in Section 2.2, it is because the online IM works are restricted among fixed number of users, while seeds in the EIM problem are selected from continuously joining users and the objective is to maximize the influence diffused to both the existing and future users. With this regard, existing online IM can be reduced as a special case of the EIM problem by simply letting the users in network remain static over time.

3.4.2 Adaption to Combinatorial Multi-Arm Bandit (CMAB). Note that the above three steps in each round naturally forms a learning-decision process, where we first learn the growing speed and activation probabilities from previous periods and then decide which users to seed for maximizing the influenced size. Based on the learning-decision process, we design a novel framework EIM to coordinate the above three steps in multiple rounds, as illustrated in Figure 1. The EIM framework converts the EIM problem into a Combinatorial Multi-Arm Bandit (CMAB) reviewed below.

Idea of CMAB. In general CMAB, there are $m$ base-arms with unknown reward distributions and, in each trial, it makes a decision to choose and trigger a super-arm that can further trigger a set of base-arms with the maximum expected rewards [38]. Then the reward obtained from each triggered base-arm is taken as the feedback to update its reward distribution, and in next trial, the decision is made under the updated reward distributions. Given the total number of trials $R$, the objective of CMAB is to design an arm selection strategy to maximize the long-term rewards obtained from the $R$ trials.

Mappings between EIM and CMAB. In the EIM framework, we model an IM campaign as one trial in the CMAB. Table 1 lists the mapping of the various components of CMAB to the EIM problem. The super-arm in each trial refers to the selected seed user set in each round, and
the base-arms correspond to the edges in network. Different from the general CMAB, here, the number of arms in the EIM framework grows with the continuously emerging new edges during network evolution. During the diffusion in the $r$th round, if the user $u_i$ gets influenced, he will try to influence all his uninfluenced neighbors joining the network until the end of the round. We take the base-arms, which correspond to the edges between $u_i$ and his uninfluenced neighbors, as the triggered base-arms. The reward of one triggered base-arm corresponding to the edge from $u_i$ to $u_j$ follows the Bernoulli distribution $\mathcal{B}(w_{ij}, r)$. Then, whether or not the activation via a triggered edge is successful or not is taken as the edge-level feedback to learn the reward distribution of the corresponding base-arm. Moreover, the reward of super-arm obtained in the $r$th trial refers to the influenced size $I(S_r, G_{r+1})$. Thus, the decision in the $r$th trial is selecting a super-arm ($S_r$) that can trigger a set of base-arms (edges) with the maximum sum of rewards ($I(S_r, G_{r+1})$). In addition, the observed new users that join the network during each round serve as the node-level feedback used to learn the growing speed of user size.

**Example.** We further give an example to facilitate the understanding of the the EIM framework. Let the budget for a viral marketing be seeding 60 users and the survival time for an IM campaign be one month. The EIM framework divides the viral marketing into multiple trials by seeding 5 users one month. Here, two consecutive trials are one month apart. Suppose that the initial network starts May 1st, and the objective of the first trial is to select 5 seeds from current users to maximize the influence among those joining before 1st of May and during 1st of May to the 31st of May. Then, the second trial is on 1st of June with the corresponding objective being maximizing the influence diffusion among the users joining until the 30th of June, and so on. In the $r$th trial, the EIM framework first learns the network knowledge from the feedbacks during previous $(r-1)$ months and then selects 5 users to maximize the influenced size during the $r$th month based on the learned growing speed and activation probabilities. The reward of the EIM framework in this example is the influenced size during the 12 months.

**Remark.** In the present work, we focus on the case where the network exhibits fast growth while the promoted information in viral marketing remains effective in a far longer period. However, we do not need to rely on any correlation between the speed of user size growing and that of influence diffusion. As long as the network is evolving, the proposed EIM framework can adaptively capture its growing speed and then selects seed users based on the learned growing speed in each trial. Even if the network is static, the EIM framework is also applicable by setting $G^{r+1} = G^r$.

### 4 LEARNING NETWORK EVOLUTION

As shown in Figure 1, the objective of the first step in the EIM framework in each round is to predict the network topology during diffusion. To this end, we require the knowledges about how
new users establish social links with existing users and how many new users will join network until the end of the round. As illustrated in Section 3.1, we adopt the well-known BA model to characterize how new users establish social links with existing users. Then, the main challenge of predicting topology lies on learning the user size growing speed \( n(t) \), which is elaborated as follows:

### 4.1 Idea of Growing Speed Learning

For learning the growing speed \( n(t) \) of a given real social network, we need to understand how the real social network size grows over time. In reality, as noted in Section 1, the network growing speed \( n(t) \) is affected by multiple factors. To elaborate, at time \( t \), the \( n(t) \) existing users prefer to attract new users to join the network, while the total population \( N \) of the users possibly joining network is limited [17]. Besides, similar to the manner of the susceptible infected (SI) model in epidemiology, users exhibit decaying interests in attracting users to join [17, 35]. Jointly considering the above factors, we adopt the Nettide-node model [17] to characterize the growing speed \( n(t) \), i.e.,

\[
d\frac{n(t)}{dt} = \beta t \cdot \theta n(t)[N - n(t)].
\]

In Equation (5), due to the limited size of potential users, growing speed is constrained by the term \([N - n(t)]\). The term \( \beta t \cdot \theta \) characterizes the decaying growing speed, and the exponent \( \theta \) reflects the growing patterns of user size in different networks, such as power law, linear, sub-linear. The Nettide-node model has been empirically justified over real social network data (e.g., Facebook, Wechat, Google-plus, and arXiv) in terms of its effectiveness in capturing networks’ growing speed, with an error of less than 3% [17]. However, the parameters (i.e., \( \beta, \theta \), and \( N \)) of a specific evolving network are unknown in advance. Thus, learning the network growing speed becomes learning the three parameters \( \beta, \theta, \) and \( N \) in Equation (5).

To this end, we borrow the idea of particle learning and propose a fully adaptive method to learn the growing speed function based on the feedback of observed network evolution. In this article, we let each particle represent a possible growing speed function with given parameters (i.e., \( \beta, \theta, \) and \( N \)) and present the definition of particles as below.

**Definition 4.1 (Particle).** In the EIM framework, each particle \( \rho_p \) represents a growing speed function with given prior parameters \( (\beta_p, \theta_p, \) and \( N_p) \), i.e.,

\[
d\frac{n_p(t)}{dt} = \beta_p t \cdot \theta_p n_p(t)[N_p - n_p(t)].
\]

Each particle has a weight \( w_p \) during the \( R \) trials.

We take the first trial as an example to briefly illustrate the process of particle learning. We assume that the topology of the initial network at timestamp \( T^1 \) is known. Let \( d_n^1 \) denote the degree of \( u_n \) at \( T^1 \). Then, under each particle \( \rho_p \), we compute the expected degree of a node \( u_n \in V^1 \) at timestamp \( T^2 \) as

\[
\mathbb{E}_p(d_n^2) = d_n^1 \cdot \prod_{s=1}^{m[n_p(T^2) - n_p(T^1)]} \left(1 + \frac{1}{\sum_{u_j \in V^1} d_j^1 + (2s - 1)} \right).
\]

Equation (6) can be obtained from Equation (2), since there are \([n_p(T^2) - n_p(T^1)]\) evolving slots during \([T^1, T^2]\) under the growing speed function \( n_p(t) \) represented by particle \( \rho_p \). Furthermore, there are some social medias (e.g., Twitter and Weibo) that can track the activities of their users such as one user retweeting a tweet forwarded by one another [10, 21]. With such fact, if a user gets influenced under the IC model, the diffusion between him and his neighbors is then observable and thus the neighbors of such user can be observed. By this, at time step \( T^2 \), if \( u_n \) gets influenced during the first trial, the real degree \( d_n^2 \) can be observed. Then, \( \mathbb{E}_p(d_n^2) \) and \( d_n^2 \), respectively, serve
as the prior and posterior values in particle learning. The weight $w_p$ of particle $\rho_p$ is set inversely proportional to the sum of $|E_p(d_n^2) - d_n^2|$ of all influenced nodes. Upon obtaining the weights, the particles with larger weights will be resampled as more new particles, while those with low weights will be killed. Through the resampling process in each trial, the remaining particles can well approximate the real growing speed $n(t)$. That is, we adopt the particle learning–based method to jointly learn the three parameters (i.e., $\beta$, $\theta$, and $N$) in the function of user size growing speed.

**Remark.** In this article, we consider a tough case that only the neighbors of the users getting influenced are observable. Notably, if the neighbors of all users are observable, the proposed growing speed learning method is also applicable. In such case, the observed degree of each node in network then serves as the posterior value in particle learning.

### 4.2 Particle Learning–based Evolution Prediction

Now, we move to the particle learning–based evolving network prediction in the $R$ trials. In each trial, we first learn the growing speed as the resampling process in particle learning and then predict the evolving network based on the resampled particles. In the $r$th trial, we use $O(T^{r})$ to denote the influenced node set that includes the nodes getting influenced during the $r$th trial. Then, the real degree of each node in $O(T^{r})$ until $T^{r+1}$ can be observed. Specially, $O(T^{0})$ includes the nodes in the initial network at $T^{1}$.

**Growing speed learning:** In the 1st trial, the particle learning is initialized by a set of particles $\mathcal{P}^{1}$ with randomly sampled prior parameters ($\beta$, $\theta$, and $N$) from their possible ranges, which will also be empirically presented in Section 8. With the progress of the EIM framework, the simulated evolving process under each particle is proceeded in parallel. In the $r$th trial, the prior value of particle $\rho_p$ on a node $u_n$ is the expected degree $\mathbb{E}_p(d_n^{r+1})$ and the posterior value is the observed degree $d_n^{r+1}$. Next, we present how to compute the value of $\mathbb{E}_p(d_n^{r+1})$.

For determining $\mathbb{E}_p(d_n^{r+1})$ under particle $\rho_p$, we divide the nodes at timestamp $T^r$ into three sets:

1. The node set $O(T^{r-1})$. The users in $O(T^{r-1})$ are those getting influenced during the $(r-1)$-th trial.
2. The node set $V^{r-1}_p\setminus O(T^{r-1})$. Given that the node size at timestamp $T^{r}$ is $n_p(T^{r})$ under the growing speed $n_p(t)$, in the simulated evolving process under particle $\rho_p$, we generate a node set $V^{r}_p$ that satisfies $|V^{r}_p| = n_p(T^{r})$ and $\bigcup_{x=0}^{r-1} O(T^{x}) \subset V^{r}_p$. That is, in such simulated evolving process, we generate a set $V^{r}_p \setminus \bigcup_{x=0}^{r-1} O(T^{x})$ of synthetic nodes to make $|V^{r}_p| = n_p(T^{r})$. Specially, we will kill the particles with $n_p(T^{r}) < |\bigcup_{x=0}^{r-1} O(T^{x})|$ in the resampling process in each trial. Then, the node set $V^{r-1}_p \setminus O(T^{r-1})$ includes the users that join the simulated network before $T^{r-1}$ but are not influenced in the $(r-1)$-th trial.
3. The node set $V^{r}_p \setminus (V^{r-1}_p \cup O(T^{r-1}))$, which includes the nodes that join during the $(r-1)$-th trial but do not get influenced during the $(r-1)$-th trial.

With the three sets, Lemma 4.2 gives the formula of $\mathbb{E}_p(d_n^{r+1})$.

**Lemma 4.2.** In the $r$th trial, the expected degree $\mathbb{E}_p(d_n^{r+1})$ under particle $\rho_p$ is given by

$$\mathbb{E}_p(d_n^{r+1}) = \bar{d}_n^{r+1} \cdot \prod_{s=1}^{m[n_p(T^{r+1})-n_p(T^{r})]} \left(1 + \frac{1}{\sum_{u_j \in V^{r}_p} \bar{d}_j^{r} + (2s-1)}\right).$$
Here, $\bar{d}'_r = \begin{cases} d'_r, u_j \in O(T^{r-1}) \\
E_p(d'_r), u_j \in V^r_p \setminus O(T^{r-1}) \\
m, u_j \in V^r_p \setminus (V^r_p \cup O(T^{r-1})) \end{cases}$ and we initially estimate the degree of a newly added node as $m$.

We present the proof for Lemma 4.2 in Appendix B in the supplementary material [42]. Furthermore, we utilize the nodes in the set $O(T^r) \cap (\bigcup_{x=0}^{r-1} O(T^x))$, which denotes the nodes get influenced not only in the $r$th trial but also in one or more of the previous $(r-1)$ trials, to compute the weights of the particles in the $r$th trial. For each node $u_n \in O(T^r) \cap (\bigcup_{x=0}^{r-1} O(T^x))$, given its last influenced time being $T^{(n,0)}$ and the corresponding observed degree being $d_n^{(n,0)}$, the prior value of particle $\rho_p$ equals
\[
\Delta n_p(T^{r+1}) = \sum_{u_n \in O(T^r) \cap (\bigcup_{x=0}^{r-1} O(T^x))} \left( E_p(d_n^{(r+1)}) - d_n^{(n,0)} \right).
\]

Meanwhile, the posterior value of the particle $\rho_p$ is given by
\[
\Delta n(T^{r+1}) = \sum_{u_n \in O(T^r) \cap (\bigcup_{x=0}^{r-1} O(T^x))} \left( d_n^{(r+1)} - d_n^{(n,0)} \right).
\]

We adopt the nodes in $O(T^r) \cap (\bigcup_{x=0}^{r-1} O(T^x))$ here because the two degrees $d_n^{r+1}$ and $d_n^{(n,0)}$ are both observed real values and we can obtain the reliable posterior values. Based on the prior and posterior values, the weight of particle $\rho_p$ is determined as $w_p(T^r) \propto 1/|\Delta n(T^{r+1}) - \Delta n_p(T^{r+1})|$. Upon obtaining the weights of particles, a resampling process is conducted to resample particles set $P^{r+1}$ from $P^r$. During the resampling process, we resample the particles with larger weights as more new particles and simultaneously kill those with low weights. Then, the remaining particles are those that better approximate the real user size growing speed. The total number of particles satisfies $|P^r| = M (1 \leq r \leq R)$.

**Evolution prediction**: The resampling process is conducted at the end of each trial. With the resampled particles in $P^{r+1}$, we then predict the topology of $G^{r+2}$ at the beginning of the $(r+1)$-th trial. Specifically, for a node $u_n$, we compute its incremental degree under particle $\rho_p \in P^{r+1}$ as
\[
E_p(\Delta d_n^{r+2}) = E_p(d_n^{(r+2)}) - d_n^{(n,0)}.
\]

Then, we set the incremental degree of node $u_n$ as the average of the estimated incremental degree under each particle, i.e., $E(\Delta d_n^{r+2}) = \sum_{p=1}^{M} \frac{1}{M} E_p(\Delta d_n^{r+2})$. As the influenced nodes in each trial and their neighbors are observed, together with the nodes in $O(T^0)$ in the initial network, all the influenced nodes in the union set $\bigcup_{x=0}^{r} O(T^x)$ and their neighbors serve as the known nodes at the beginning of the $(r+1)$-th trial. We use the set $K(T^{r+1})$ to denote the set of known users until $T^{(r+1)}$. Further, we conduct the evolution prediction through predicting the incremental degrees of such known nodes in each trial. Such incremental degrees then quantify how many unknown users that each known user possibly further influences after him getting influenced during diffusion. Specifically, for the known users that have never been influenced, we set $d_n^{(n,0)} = 0$.

The pseudo code of the above growing speed learning and evolution prediction process is summarized in Algorithm 1 called Evo-NE. At the end of the $r$th trial, Evo-NE resamples the particles based on their weights determined from the prior value $\Delta n_p(T^{r+1})$ and the posterior value $\Delta n(T^{r+1})$. Then, at the beginning of the $(r+1)$-th trial, Evo-NE computes the incremental degrees of the known nodes in $K(T^{r+1})$ in the evolution prediction. Since Evo-NE needs to traverse the known node set for each particle, the complexity of Evo-NE scales as $O(M |V'|)$ in the $r$th trial.
5 LEARNING EVOLVING ACTIVATION PROBABILITIES

In Section 4, we have illustrated the proposed method for learning the growing speed and predicting network evolution, which is the step (1) in the EIM framework. With the evolving network characterizing how the seed set triggers the base-arms, the evolving activation probabilities then specify the rewards obtained from the triggered base-arms and are also required for identifying the seed set to maximize the reward $I(S_r, G^{r+1})$. In this section, we present our method for learning activation probabilities, which is the step (2) in the EIM framework.

Idea of learning evolving activation probabilities. During the diffusion in the $r$th trial, if user $u_i$ gets influenced, he will try to influence his neighbor $u_j$ successfully with probability $w_{ij,r}$. Let the binary variable $z_{ij,r}$ denote whether $u_i$ influences $u_j$ successfully ($z_{ij,r} = 1$) or not ($z_{ij,r} = 0$), $z_{ij,r}$ then follows the Bernoulli distribution $B(w_{ij,r})$. By this, we adopt the edge-level feedback $z_{ij,r}$ to estimate the evolving activation probability via the edge between $u_i$ and $u_j$ at the end of the $r$th trial. In this article, we assume that the activating results (e.g., $z_{ij,r}$) via edges are observable. Our insight for such assumption is that some social medias (e.g., Twitter and Weibo) can track the user activities that one user retweets a tweet forwarded by his neighbor [10, 21]. If $u_j$ reweets a tweet from $u_i$ during the $r$th round, we say $u_i$ successfully influences $u_j$ and tag $z_{ij,r} = 1$.

In detail, recalling the evolving model of activation probabilities introduced in Equation (3), the value of $w_{ij,r}$ forms a Gaussian random walk. In the $r$th trial, we use $\bar{w}_{ij,r}$ to denote the expectation of $w_{ij,r}$ and use variance $\Sigma_{ij,r}$ to characterize the uncertainty of $w_{ij,r}$. Notably, if we do not refine the expectation based on the edge-level feedback, according to Equation (3), there is $\bar{w}_{ij,r} = \bar{w}_0$ and $\Sigma_{ij,r} = \Sigma_0 + \Delta \Sigma_{ij,x}$. Thus, our objective is utilizing the edge-level feedback to obtain the more reliable expectation $\bar{w}_{ij,r}$. With the Gaussian statistical property of $w_{ij,r}$, we adopt the Kalman Filter to refine the expectation $\bar{w}_{ij,r}$ and the corresponding variance $\Sigma_{ij,r}$.

Kalman filter–based refining method. Given the edge-level feedback $z_{ij,r}$, referring to the Kalman filter theory [37], at the end of the $r$th trial, the expectation $\bar{w}_{ij,r}$ and the variance $\Sigma_{ij,r}$ of the activating probability $w_{ij,r}$ are refined as

$$\bar{w}_{ij,r+1} = \bar{w}_{ij,r} + G_{ij,r} \cdot (z_{ij,r} - \bar{w}_{ij,r}),$$
$$\Sigma_{ij,r+1} = \Sigma_{ij,r} (1 - G_{ij,r}) + \Delta \Sigma_{ij,r+1},$$

where $G_{ij,r}$ is the Kalman gain, $\Delta \Sigma_{ij,x}$ is the uncertainty of $\bar{w}_{ij,r}$, and $\Delta \Sigma_{ij,r+1}$ is the uncertainty of $\bar{w}_{ij,r+1}$. With these equations, we can obtain the refined expectations for the $(r+1)$th trial.
Here, $G_{ij,r}(z_{ij,r} - \overline{w}_{ij,r})$ and $G_{ij,r} \Sigma_{ij,r}$ are the refinement from Kalman filter, and $G_{ij,r}$ is the Kalman Gain that quantifies the refinement from the new observation $z_{ij,r}$. Lemma 5.1 presents the formula of $G_{ij,r}$ in the EIM framework.

**Lemma 5.1.** The Kalman Gain in the refinement of the distribution of $w_{ij,r}$ at the end of the $r$th trial is determined by $G_{ij,r} = \Sigma_{ij,r} \cdot Q_{ij,r}^{-1}$, where $Q_{ij,r} = \Sigma_{ij,r} + 1$ bounds the variance of the edge-level feedback $z_{ij,r}$.

The proof for Lemma 5.1 is deferred to Appendix C in the supplementary material [42]. However, if the edge from $u_i$ to $u_j$ is not triggered in the $r$th trial, the distribution of the activation probability evolves as

$$ w'_{ij,r+1} = \overline{w}_{ij,r}, \Sigma_{ij,r+1} = \Sigma_{ij,r} + \Delta \Sigma_{ij,r+1}. \tag{12} $$

Combining the two cases of both triggered and non-triggered edges, Algorithm 2 named **Evo-IL** gives the pseudo code for the evolving activation probability learning at the end of the $r$th trial. **Evo-IL** takes the edge-level feedback via each triggered edge as the input and outputs the updated expectations and variances of the activation probabilities via the edges of all influenced nodes. In the $r$th trial, **Evo-IL** needs to traverse both the triggered and non-triggered edges and costs a complexity of $O(|E'|)$.

**Algorithm 2:** Evolving activation probability learning (**Evo-IL**)

```plaintext
// Activation probability learning at the end of the $r$th trial
Input: Edge-level feedback during $T'$ to $T'+1$;
Output: $w'_{ij,r+1}, \Sigma_{ij,r+1}$ of the edges of nodes in $\bigcup_{x=0}^{r} O(T')$;
1 Set $\overline{w}_{ij,r} = w_0$ and $\Sigma_{ij,r} = \Sigma_0$ for each observed newly established edges;
2 for each triggered edge do
3     Compute $Q_{ij,r} = \Sigma_{ij,r} + 1$;
4     Compute $G_{ij,r} = \Sigma_{ij,r} \cdot Q_{ij,r}^{-1}$;
5     Update $\overline{w}_{ij,r+1} = \overline{w}_{ij,r} + G_{ij,r} \cdot (z_{ij,r} - \overline{w}_{ij,r})$;
6     Update $\Sigma_{ij,r+1} = \Sigma_{ij,r}(1 - G_{ij,r}) + \Delta \Sigma_{ij,r+1}$;
7 end
8 for each non-triggered edge do
9     Update $\overline{w}_{ij,r+1} = \overline{w}_{ij,r}$;
10  Update $\Sigma_{ij,r+1} = \Sigma_{ij,r} + \Delta \Sigma_{ij,r+1}$;
11 end
12 return $\overline{w}_{ij,r+1}, \Sigma_{ij,r+1}$ for each edge.
```

**Upper confidence bound–based estimator.** Based on the expectation $\overline{w}_{ij,r+1}$ and the variance $\Sigma_{ij,r+1}$, at the beginning of the $(r + 1)$-th trial, we adopt the Upper Confidence Bound (UCB) method to estimate the activation probability $w'_{ij,r+1}$, which we use in the seed selection in the $(r + 1)$-th trial. In the traditional UCB framework [21], given the expectation of the reward $A$ of an arm, and its variance $\Sigma$, the estimator of the reward is determined by $A' = A + c\sqrt{\Sigma}$. Accordingly, the estimated value $w'_{ij,r+1}$ is expressed as Definition 5.2.

**Definition 5.2.** The estimator $w'_{ij,r+1}$ at the beginning of the $(r + 1)$-th trial is:

$$ w'_{ij,r+1} = \overline{w}_{ij,r+1} + c\sqrt{\Sigma_{ij,r+1}}, \tag{13} $$

where $c$ is a constant algorithm parameter in the Linear generalization of UCB (LinUCB) [21].

**Remark.** We adopt the UCB method here for balancing the tradeoff between the exploitation and exploration in seed selection. As shown in Definition 5.2, for exploitation, we estimate $w'_{ij,r+1}$
based on the current expectation $\overline{w}_{ij,r+1}$. For exploration, we add the term $c\sqrt{\Sigma_{ij,r+1}}$ into $w'_{ij,r+1}$. This is because the variance $\Sigma_{ij,r+1}$, as well as the term $c\sqrt{\Sigma_{ij,r+1}}$, decreases with the triggered times of the corresponding edge, and the edges being triggered for fewer times have comparatively larger values of $w'_{ij,r+1}$. Moreover, the larger estimated activation probability means more expected reward if the corresponding base-arm is being triggered. Thus, as the aim of seed selection is maximizing the rewards of triggered edges, the term $c\sqrt{\Sigma_{ij,r+1}}$ motivates the EIM framework to select the seeds that could then trigger the edges with fewer triggered times and to obtain more reliable expectations of the activation probabilities via such edges.

6 EVOLVING SEED SELECTION

With Definition 5.2 giving the estimators of activation probabilities at the beginning of each trial, we then adopt such estimators (e.g., $w'_{ij,r}$) in seed selection. Recalling Figure 1, seed selection is the step (3) in the EIM framework with the main task being, based on the predicted network evolution (Section 4) and the estimators of activation probabilities (Section 5), selecting a set of seeds to maximize the reward $I(S_r, G^{r+1})$.

6.1 Evolving Seed Selection: Problem Transformation

As stated earlier, the seed selection in classical IM and the EIM problems are both NP-hard, and there have been a plenty of works on solving the classical IM problem. Our design in the evolving seed selection for the EIM problem is not to jettison the previous efforts in classical IM, but instead to leverage the benefits of them wherever possible. Comparing with the classical IM problem, the major challenge for the EIM problem comes from the unknown network evolution and activation probabilities. In Sections 4 and 5, we have presented our solution for predicting network evolution, which is characterized by the expected incremental degree $E(\Delta d^r_{i,j})$ of known users, and the solution for learning the evolving activation probabilities, which are characterized by the estimators (e.g., $w'_{ij,r}$). Then, our idea of designing the evolving seed selection for maximizing the influenced size $I(S_r, G^{r+1})$ is applying the classical IM solutions onto the predicted network evolution and the estimators of activation probabilities. To this end, we first generate an intermediate graph $G^r$ to capture the future network $G^{r+1}$.

Intermediate Evolving Graph $G^r = (\mathbb{V}^r, \mathbb{E}^r)$. As the seeds in the $r$th trial are selected from the known nodes in $K(T^r)$, we let $\mathbb{V}^r = K(T^r)$ and let $\mathbb{E}^r$ consist of the observed edges among nodes in $\mathbb{V}^r$. The activation probability via the edge from $u_i$ to $u_j$ over $G^r$ is equal to the estimator $w'_{ij,r}$. Then, each node $u_i \in \mathbb{V}^r$ has a weight $C_{i,r}$ that is equal to the expected incremental degree $E(\Delta d^r_{i})$. The weight $C_{i,r}$ serves as the index for the size of the potential users possibly being influenced by $u_i$ until time $T^{r+1}$. Here, the potential users refer to the future neighbors joining network during $[T^r, T^{r+1})$ and those joining before $T^r$ but having never been observed. Since the potential users include the future neighbors, we also adopt the weight $C_{i,r}$ to quantify the attractiveness of $u_i$ to future users.

Over the intermediate evolving graph $G^r$, if a user with higher weight gets influenced during diffusion under the IC model starting from the seed set $S_r$, such user can further influence more potential users. Then, with the aim of maximizing $I(S_r, G^{r+1})$, each user cannot be equally treated in seed selection. Thus, over the intermediate evolving graph $G^r$, we transform the EIM problem in each trial in Equation (4) as follows:

Problem transformation. Let $I(S_r, u_i, G^r)$ denote the probability that seed set $S_r$ can influence $u_i$ over $G^r$. As the weight of each user $u_i$ is $C_{i,r}$, we quantify the expected reward of influencing $u_i$ as $I(S_r, u_i, G^r)C_{i,r}$ and quantify the expected reward obtained from the seed set $S_r$ as the weighted sum $I(S_r, G^r) = \sum_{u_i \in \mathbb{V}^r} I(S_r, u_i, G^r)C_{i,r}$. Thus, we transform the EIM problem in Definition 3.1
into selecting a seed set
\[
\hat{S}_{\text{opt}} = \arg\max_{S_r \subseteq \mathcal{V}_r, |S_r| = K} I(S_r, G^r) \quad (1 \leq r \leq R),
\] (14)
in the seed selection in each round.

6.2 Evolving Seeds Selection: Algorithm Design

Over the intermediate evolving graph \( G^r \), we leverage the Influence Maximization via Martingale (IMM) framework to solve the EIM problem transformed into Equation (14). The IMM framework focuses on estimating the influence diffusion size under IC model via the Reverse-Reachable Sets (RR-sets). The RR-sets [8] is currently the most efficient way to estimate the influence diffusion size under IC model with provable high accuracy and has been adopted by many IM techniques.

**RR-sets.** For the classical IM problem over a static network \( G \), the RR-sets are generated as follows: A deterministic copy \( g \) of \( G \) is first sampled, where each edge \( e \) in \( G \) is tagged as active with the activation probability \( w_e \) via it and as inactive with probability \( 1 - w_e \). Then a node \( v \) in \( G \) is randomly chosen, and a reverse Breadth-First-Search (BFS) process starting from \( v \) is conducted over the copy \( g \). Furthermore, the RR-set \( R_v \) for node \( v \) includes all the nodes that the reverse BFS can reach, and \( v \) is called as the root node of \( R_v \). The rationale of the RR-set is that the nodes in \( R_v \) are those who probably influence node \( v \) during diffusion. An RR-set is taken as a sample of the influence diffusion process under IC model.

**General IMM.** Based on the RR-sets, the general IMM framework consists of two phases, i.e., sampling and seed selection. The former phase first determines the required number of RR-sets for guaranteeing the accuracy of influence estimation and then generates the determined number of RR-sets. We say a node covers an RR-set if the node belongs to the RR-set. According to the Chernoff Bound, under a sufficiently large number of random RR-sets, a node covering a larger number of RR-sets can expectedly influence more users during diffusion [8]. Thus, the second phase of IMM is greedily selecting a set of nodes who can maximize the number of covered RR-sets as the seeds. Next, borrowing the idea of IMM, our evolving seed selection algorithm \textbf{Evo-IMM} is as follows:

**ALGORITHM 3:** Evolving influence maximization (Evo-IMM).

\begin{algorithm}
\begin{algorithmic}[1]
\Input {Intermediate evolving graph \( G^r \), number of seeds \( K \);}
\Output {A seed set \( S_r \);}
1 \textbf{R} = Sampling (\( G^r \), \( K \), \( \epsilon \), \( I \));
2 \textbf{S}_r = Seed Selection (\( R \), \( K \));
3 \Return \textbf{S}_r.
\end{algorithmic}
\end{algorithm}

**Evolving IMM algorithm (Evo-IMM).** Since each node in \( G^r \) is attached with a weight, each RR-set sampled from \( G^r \) is also attached with a weight of its root node. We call such RR-sets as ERR-sets. Algorithm 3 gives the basic steps of \textbf{Evo-IMM} in the \( r \)th trial, in a similar manner to the general IMM. However, as the nodes in \( G^r \) have different weights, in the Sampling (\( G^r \), \( K \), \( \epsilon \), \( I \)) (Algorithm 4), we propose to determine the required weighted sum of generated ERR-sets to guarantee the quality of selected seeds and then iteratively generate a set \( \hat{R} \) of ERR-sets until reaching the required weighted sum \( \theta \). The \( \epsilon \) and \( I \) are the parameters for balancing the tradeoff between performance guarantee and the time complexity of \textbf{Evo-IMM}. The detail of determining the required weighted sum \( \theta \) is in Appendix D in the supplementary material [42]. Notably, when generating the ERR-sets, we propose a priority-based sampling method to preferentially sample the nodes with high weights as the root nodes. In Algorithm 4, the nodes in \( \mathcal{V}^r \setminus R_{\text{root}} \) with higher weights have the higher probabilities to be chosen as the next root node. Accordingly, in the Seed
Selection phase (Algorithm 5), we focus on greedily selecting the seed set with the maximum sum of the weights of covered ERR-sets. Let \( F_R(\cdot) \) denote the weighted sum of covered ERR-sets, Seed Selection iteratively selects the node with the maximum marginal gain to \( F_R(S_r) \).

**Algorithm 4:** Sampling \((G', K, \epsilon, l)\)

**Input:** \( G', K, \epsilon, l \);
**Output:** RR-sets \( \mathcal{R} \);
1. \( \theta = \) Determining the required weighted sum of ERR-sets based on \( G', K, \epsilon, l \);
2. Initialize \( n' = 0, \theta' = 0, \mathcal{R} = \emptyset \) and \( \mathcal{R}_{\text{root}} = \emptyset \);
3. for each \( u_e \in \mathcal{V}' \) do
   4. \( \lambda_e = C_{e, r}, n' = n' + C_{e, r} \);
4. end
5. Divide interval \([0, n']\) into \([0, \lambda_1], [\lambda_1, \lambda_1 + \lambda_2], \ldots, [\sum_{j=1}^{|\mathcal{V}'| - 1} \lambda_j, n']\);
6. while \( \theta' < \theta \) do
   7. Randomly sample a constant \( \alpha \) from interval \([0, 1]\);
   8. \( n_\alpha = n' \cdot \alpha \);
   9. if \( \sum_{j=0}^{\epsilon-1} \lambda_j \leq n_\alpha \leq \sum_{j=0}^{\epsilon} \lambda_j \) and \( u_e \notin \mathcal{R}_{\text{root}} \) then
   10. \( u = u_e \);
   11. else Continue;
   12. end
   13. \( \mathcal{R}_{\text{root}} = \mathcal{R}_{\text{root}} \cup v \);
   14. \( n' = n' - C_{e, r}, \theta' = \theta' + C_{e, r} \);
   15. Generating RR-set \( \mathcal{R}_v \) starting from node \( v \);
   16. \( \mathcal{R} = \mathcal{R} \cup \mathcal{R}_v \);
17. end
18. return RR-sets \( \mathcal{R} \).

**Algorithm 5:** Seed Selection \((\mathcal{R}, K)\)

**Input:** ERR-sets \( \mathcal{R} \), number of seeds \( K \);
**Output:** A seed set \( S_r \);
1. Initialize a seed set \( S_r = \emptyset \);
2. for \( k = 1 : K \) do
   3. \( u_e = \arg \max_{u_e \in \mathcal{V}' \setminus S_r} F_R(S_r \cup u) - F_R(S_r) \);
   4. \( S_r = S_r \cup \{u_e\} \);
5. end
6. return \( S_r \).

Theorem 6.1 presents the performance guarantee and the time complexity of the **Evo-IMM** algorithm. The proof of Theorem 6.1 is given in Appendix D in the supplementary material [42].

**Theorem 6.1.** **Evo-IMM** returns a \((1 - 1/e - \epsilon)\)-approximate seed set to Equation (14) with a probability of at least \((1 - 1/n) (n \geq 1)\), and has a complexity of \( O((K + l)(|\mathcal{V}'| + |E'|)(\log |\mathcal{V}'|)/\epsilon^2) \).

**6.3 A Summary of Sections 4, 5, and 6**

Recalling Section 3, the major challenges of the **EIM** problem in Equation (4) come from the unknown network topology and activation probabilities. The main tasks in Sections 4 and 5 are learning the network evolution and the activation probabilities, which enable us to design the seed selection algorithm in Section 6 based on classical IM solutions. Thus, the three steps of the **EIM** framework in each trial as shown in Figure 1 are algorithmically conducted by Algorithms 1, 2, and 3. Together with Algorithms 1, 2, and 3, the complexity of the \( r \)th trial in the **EIM**
framework is \( O(M|V'| + |E'| + (K + I)(|V'| + |E'|) \log |V'|/\varepsilon^2) \). Here, \( M \) is the number of particles in Algorithm 1.

Notably, in Theorem 6.1, we just give the performance guarantee of the returned seed set \( S_r \) to the transformed problem in Equation (14). Considering the EIM problem in Equation (4), a natural question is how is the performance of the seed set \( S_r \) to the EIM problem? In the sequel, we answer this question via the performance analysis of the \( \text{EIM} \) framework.

7 PERFORMANCE ANALYSIS OF \( \text{EIM} \)

Recall again that in Section 3.4, \( \text{EIM} \) is a bandit-based framework where the base-arms represent the edges in network and the super-arm is the seed set. Then, to demonstrate the theoretical performance guarantee of the \( \text{EIM} \) framework, we provide the analysis for its Regret, which is quantified by the gap between the reward of the selected super-arm and the reward of the optimal super-arm. In the \( \text{EIM} \) problem, the reward refers to the influence diffusion size, and, intuitively, the regret of the \( \text{EIM} \) framework in the \( r \)th trial can be given by \( I(S_r^{opt}, G^{r+1}) - I(S_r, G^{r+1}) \). Here, \( S_r^{opt} \) denotes the optimal seed set in the \( r \)th round, and \( S_r \) is the seed set returned by the \( \text{EIM} \) framework. However, recalling Lemma 3.2, the \( \text{EIM} \) problem is NP-hard. As a result, even if the network \( G^{r+1} \) is fully known in advance, the selected seed set can only be a suboptimal one and achieve an expected influenced size of \((1 - 1/e - \varepsilon) I(S_r^{opt}, G^{r+1}) \) with a probability of at least \((1 - 1/n')\). Then, since \( S_r \) is actually returned by an approximate seed selection algorithm over the estimated network, the regret \( I(S_r^{opt}, G^{r+1}) - I(S_r, G^{r+1}) \) is caused by two factors, i.e., unknown network knowledge and approximate seed selection algorithm. By this, for evaluating the performance of \( \text{EIM} \) framework on coping with the unknown network knowledge, we adopt the scaled cumulative regret, which is a widely used metric for the approximation bandit [2, 21, 38], to evaluate the performance of the \( \text{EIM} \) framework.

Definition 7.1 (Scaled Cumulative Regret). Given the approximating ratio of IM algorithm in step (3) being \( \beta \), the cumulative regret \( B \) of the \( \text{EIM} \) framework over \( R \) trials is equal to

\[
\mathbb{E}(B) = \sum_{r=1}^{R} \mathbb{E} \left( I(S_r^{opt}, G^{r+1}) - \frac{1}{\beta} I(S_r, G^{r+1}) \right),
\]

(15)

where \( \beta = (1 - 1/e - \varepsilon) \cdot (1 - 1/n') \) for the \( \text{EIM} \) framework, and the parameter \( \frac{1}{\beta} \) serves as the compensation for the partial regret caused by the approximate seed selection algorithm.

Under Definition 7.1, the upper bound of the scaled regret of the \( \text{EIM} \) framework is disclosed in Theorem 7.2.

Theorem 7.2. The regret of the \( \text{EIM} \) framework is upper bounded by

\[
\mathbb{E}(B) \leq O \left( \sqrt{|E^{R+1}| \ln(R + 1)R} \right).
\]

(16)

Here, \(|E^{R+1}|\) denotes the number of edges until time \( T^{R+1} \), when the \( k \) in Equation (3) satisfies \( k \geq 2 \).

Proof. We divide the whole proof into four steps.

1. Overall regret over \( R \) trials. In the \( r \)th trial, we, respectively, incorporate the real activation probabilities \( w_{ij,r} \) and the estimated activation probabilities \( w'_{ij,r} \) via edges into the vectors \( \hat{w}_r \) and \( \hat{w}'_r \). Then, \( \mathbb{E} \left[ I(S, \hat{w}_r) \right] \) and \( \mathbb{E} \left[ I(S, \hat{w}'_r) \right] \) denote the expected influenced size of a seed set \( S \) under \( \hat{w}_r \) and \( \hat{w}'_r \), respectively. Throughout the proof, we use \( w_{e,r} \) to simplify the notation \( w_{ij,r} \). Recalling the estimator \( w_{e,r} \) of activation probability in Definition 5.2 (i.e., \( w_{e,r}' = \hat{w}_{e,r} + c \frac{1}{\Sigma_{e,r}} \)),
we define an event $\mathcal{F}_r$ to bound the estimating error of the expectation $\overline{w}_{e,r}^\prime$, i.e.,

$$\mathcal{F}_r \triangleq |\overline{w}_{e,r}^\prime - w_{e,r}| \leq c\sqrt{\Sigma_{e,r}}, \forall e \in E^{r+1}.$$ 

With event $\mathcal{F}_r$, we rewrite the expected regret according to the law of total expectation as

$$\mathbb{E}[B] = \sum_{r=1}^{R} \mathbb{E}
\left[
\left| I(S_r^{opt}, \overline{w}_r) - \frac{1}{\beta} I(S_r, \overline{w}_r) \right| \right] P(\mathcal{F}_r) + \sum_{r=1}^{R} \mathbb{E}
\left[
\left| I(S_r^{opt}, \overline{w}_r) - \frac{1}{\beta} I(S_r, \overline{w}_r) \right| \mathbb{F}_r \right] P(\overline{\mathcal{F}}_r).
$$

(17)

Furthermore, under the event $\mathcal{F}_r$, we have $0 \leq w_{e,r}^\prime - w_{e,r} \leq 2c\sqrt{\Sigma_{e,r}}$ holds for all edges, and due to $w_{e,r}^\prime \geq w_{e,r}$, we further have

$$I(S_r^{opt}, \overline{w}_r) \leq I(S_r^{opt}, \overline{w}_r^\prime) \leq \frac{1}{\beta} \mathbb{E}[I(S_r, \overline{w}_r)].$$

Then, taking the above inequality into Equation (17), we obtain an upper bound of the regret that

$$\mathbb{E}[B] \leq \sum_{r=1}^{R} \mathbb{E}
\left[
\frac{1}{\beta} I(S_r, \overline{w}_r^\prime) - \frac{1}{\beta} I(S_r, \overline{w}_r) \mathbb{F}_r \right] P(\mathcal{F}_r) + \sum_{r=1}^{R} \mathbb{E}
\left[
\left| I(S_r^{opt}, \overline{w}_r) - \frac{1}{\beta} I(S_r, \overline{w}_r) \right| \mathcal{F}_r \right] P(\overline{\mathcal{F}}_r)

\leq \sum_{r=1}^{R} \frac{1}{\beta} \mathbb{E}
\left[
|I(S_r, \overline{w}_r^\prime) - I(S_r, \overline{w}_r)| \mathcal{F}_r \right] + \sum_{r=1}^{R} P(\overline{\mathcal{F}}_r)|V^{r+1}|.
$$

(18)

Here, $P(\overline{\mathcal{F}}_r)|V^{r+1}|$ means that the regret is no more than $|V^{r+1}|$ even under the worst case. Next, we continue to, respectively, derive the upper bounds of $L1$ and $L2$.

2. **Upper bound of $L1 = \sum_{r=1}^{R} \frac{1}{\beta} \mathbb{E}[I(S_r, \overline{w}_r^\prime) - I(S_r, \overline{w}_r)|\mathcal{F}_r]$.** To facilitate the computation of $I(S_r, \overline{w}_r^\prime)$ and $I(S_r, \overline{w}_r)$, we approximate the evolving network as evolving forest where we only take account of one diffusion path between a pair of users. The reason for adopting the forest model for approximation is as follows: Under the IC model, the influences diffused from seeds to other nodes are expectedly along the path with the maximum activation probability [21, 39]. As the influence diffusion is progressive, in each trial, if a node has been influenced by the seeds through the path with the maximum activation probability, then such node will remain influenced and will not be influenced again. Thus, the forest that only takes account of one path between each pair of nodes is widely utilized in existing works for computing the expected influences of seeds, with the most representative one being the Maximum Influence Arborecence (MIA) model [21, 39].

By modeling evolving network as the evolving forest, we only count one path from any seed in $S_r$ to the nodes in $V^{r+1}\setminus S_r$. Next, we move to quantify the values of $\mathbb{E}[I(S_r, \overline{w}_r^\prime)]$ and $\mathbb{E}[I(S_r, \overline{w}_r)]$ based on the forest. Let $T_{r,v}$ denote the set of diffusion paths from seeds in $S_r$ to the node $v \in V^{r+1}\setminus S_r$ under IC model. Then, in the forest model, the activation probability of a diffusion path equals the product of the activation probabilities via the edges that make up the path. Further, we prove in the Appendix E in the supplementary material [42] that the value of $L1$ can be upper bounded by a polynomial of the estimating errors of activation probabilities (i.e., $(w_{e,r}^\prime - w_{e,r})$). That is,

$$L1 = \sum_{r=1}^{R} \frac{1}{\beta} \mathbb{E}
\left[
|I(S_r, \overline{w}_r^\prime) - I(S_r, \overline{w}_r)| \mathcal{F}_r \right] \leq \frac{1}{\beta} \sum_{r=1}^{R} \sum_{v \in V^{r+1}\setminus S_r} \sum_{e \in T_{r,v}} \mathbb{E}
\left[
\left| o_e^r(w_{e,r}^\prime - w_{e,r}) \right| \right].
$$

(19)
In Inequality (19), \( \mathbb{I}(o_e^r) \) denotes the event that edge \( e \) is triggered in the \( r \)th trial. We use \( P(o_e^r) \) to denote the probability of event \( \mathbb{I}(o_e^r) \) with \( P(o_e^r) = \mathbb{E}[\mathbb{I}(o_e^r)] \). Since \( 0 \leq w_{e,r} - w_{e,r} \leq 2e^2 \Sigma_{e,r} \) under the event \( \mathcal{F}_r \), Inequality (19) further becomes

\[
L_1 \leq \sum_{r=1}^{R} \frac{2c}{\beta} \sum_{e \in E^{r+1}} \mathbb{E} \left[ \mathbb{I}(o_e^r) N_{r,e} \Sigma_{e,r} \right].
\]

Here, \( N_{r,e} \) is the number of the paths in \( \bigcup_{v \in V^{r+1}\setminus S_r} \mathcal{T}_{r,v} \) that contain edge \( e \). Then, according to the Cauchy-Schwarz Inequality, we have

\[
\mathbb{E} \left[ \sum_{r=1}^{R} \sum_{e \in E^{r+1}} \mathbb{I}(o_e^r) N_{r,e} \Sigma_{e,r} \right] \leq \sqrt{\mathbb{E} \left[ \sum_{r=1}^{R} \sum_{e \in E^{r+1}} \mathbb{I}(o_e^r) N_{r,e}^2 \right] \mathbb{E} \left[ \sum_{r=1}^{R} \sum_{e \in E^{r+1}} \mathbb{I}(o_e^r) \Sigma_{e,r} \right]}.
\]

Since the number of edges grows with network evolution, we define a network parameter \( C = \max_{S_r:|S_r|=K,1 \leq r \leq R} \sqrt{\sum_{e \in E^{r+1}} N_{e,r}^2 \cdot P(o_e^r)} \) to bound the effect of network size on the regret. Such parameter \( C \) enables us to explore the correlation between the regret of the EIM framework and the number of trials.

With parameter \( C \), the Inequality (20) becomes

\[
\mathbb{E} \left[ \sum_{r=1}^{R} \sum_{e \in E^{r+1}} \mathbb{I}(o_e^r) N_{r,e} \Sigma_{e,r} \right] \leq C \sqrt{R} \sqrt{\sum_{r=1}^{R} \sum_{e \in E^{r+1}} \mathbb{I}(o_e^r) \Sigma_{e,r}}.
\]  

To give the upper bound of Equation (21), next, we provide the analysis of the term \( \sum_{r=1}^{R} \mathbb{I}(o_e^r) \Sigma_{e,r} \).

**Upper bound of \( \sum_{r=1}^{R} \mathbb{I}(o_e^r) \Sigma_{e,r} \) in Equation (21).**

1° We first consider a special case where an edge is triggered in all the \( R \) trials. Without loss of generality, we take edge \( e \) as an example and use \( \Sigma_r \) to denote its variance in the \( r \)th trial in this analysis of \( \sum_{r=1}^{R} \mathbb{I}(o_e^r) \Sigma_{e,r} \). In this special case, the edge \( e \) is established in the initial network with \( \Delta \Sigma_{e,r} = \frac{2e^2}{r^2} \). By Equation (11), we have

\[
\Sigma_{r+1} \leq \frac{\Sigma_0}{(r+1)^k} + \frac{\Sigma_r}{\Sigma_r + 1}.
\]

Since \( 0 \leq w_r \leq 1 \), we let \( \Sigma_1 \leq 1 \leq \frac{3}{r^\frac{1}{2}} \) and explore the upper bound of \( \Sigma_r \) by induction. Concretely, if \( \Sigma_{r-1} \leq \frac{3}{(r-1)^\frac{1}{2}} \) and \( k \leq 2 \), then

\[
\Sigma_r \leq \Delta \Sigma_r + \frac{3}{(r-1)^\frac{1}{2}} \frac{1}{r^k} + \frac{3}{(r-1)^\frac{1}{2}} + 3 \leq \frac{3}{r^\frac{1}{2}}.
\]

The proof for Inequality (23) is in Appendix E in the supplementary material [42]. Hence, by induction, we can draw the conclusion that \( \Sigma_r \leq \frac{3}{r^\frac{1}{2}} \). Furthermore, when \( 0 < k < 2 \), we have

\[
\sum_{r=1}^{R} \Sigma_r \leq \sum_{r=1}^{R} \frac{3}{r^k} \leq \frac{6}{2-k} R^{1-k} (0 < k < 2).
\]

The proof for Inequality (24) is in Appendix E in the supplementary material [42]. For \( k \geq 2 \), we have \( \Sigma_r \leq \frac{3}{r} \), and \( \sum_{r=1}^{R} \Sigma_r \) becomes

\[
\sum_{r=1}^{R} \Sigma_r \leq \sum_{r=1}^{R} \frac{3}{r} \leq 3 \ln T + 3.
\]
2. In Appendix E in the supplementary material [42], we further prove that, in general cases where edge $e$ is not triggered in one or more trials, $\sum_{r=1}^{R} (s_e^r) \Sigma_{r,e} \leq \sum_{r=1}^{R} \frac{3}{\sqrt{R}}$ still holds.

Taking the upper bound of $\sum_{r=1}^{R} (s_e^r) \Sigma_{r,e}$ into Inequality (21), we then have $L1 \leq O\left(\sqrt{|E_{R+1}|} \sum_{r=1}^{R} \frac{3}{\sqrt{R}} \right) = O(\sqrt{|E_{R+1}|} \ln(R + 1)R)$.

3. The upper bound of $L2 = \sum_{r=1}^{R} P(\overline{F}_r)|V^{r+1}|$. We first review the definition of event $\overline{F}_r$, i.e., $\overline{F}_r \triangleq \{|\overline{w}_{e,r} - w_{e,r}| \leq c \sqrt{\Sigma_{e,r}} \land \forall e \in E^{r+1}\}$. Next, we explore the upper bound of $L2$ based on the property of the sub-Gaussian distribution, which is a tail decaying probability distribution and contains the widely known Gaussian distribution.

Recalling the definition in Equation (3), $w_{e,r}$ follows the Gaussian distribution and thus follows the sub-Gaussian distribution. Then, $w_{e,r}'$ is the estimating value of $w_{e,r}$, by Corollary 1.7 in Reference [40], $(w_{e,r} - w_{e,r})'$ is a sub-Gaussian random variable with the variance upper bounded by $\Sigma_{e,r}$. According to the Lemma 1.3 in Reference [40], we have

$$P(|w_{e,r} - w_{e,r}'| > c\sqrt{\Sigma_{e,r}}) \leq 2e^{-\frac{c^2}{2}}.$$  

Let $|\overline{E}|$ denote the average edge size in the $R$ trials, then

$$\sum_{r=1}^{R} P(\overline{F}_r)|V^{r+1}| \leq 2e^{-\frac{c^2}{2}} \cdot |\overline{E}| |V^{R+1}|R.$$  

In the case that $c \geq \sqrt{2\ln 2|\overline{E}| |V^{R+1}| R}$, we have $\sum_{r=1}^{R} P(\overline{F}_r)|V^{r+1}| \leq 1$.

4. Conclusion. Together with the upper bounds of both $L1$ and $L2$, we can derive the regret bound of the EIM framework over the $R$ trials. Let $c = 2\sqrt{\ln 2|\overline{E}| |V_{R+1}| R}$, according to Equation (21) and Equation (24), for $0 < k < 2$, we have

$$\mathbb{E}[B] \leq \frac{2cC}{\beta} \sqrt{\frac{6}{2-k} |E_{R+1}| R^{1-\frac{k}{2}}} + 1. \quad (26)$$  

Moreover, referring to Equation (25), for $k \geq 2$, we have

$$\mathbb{E}[B] \leq \frac{2cC}{\beta} \sqrt{3 |E_{R+1}| (\ln R + 1)R} + 1 = O(\sqrt{|E_{R+1}| (\ln R + 1)R}).$$  

Thus, we complete the proof for Theorem 7.2. \hfill \Box

Theorem 7.2 implies that the regret bound of the EIM framework is still sublinear to the number of trials under the growing network size. The sub-linearity of the regret bound justifies that the EIM framework can effectively learn the knowledge of the evolving network and achieve the long-run reward that converges to the reward of the case of knowing full network knowledge. In Section 8, we will further experimentally demonstrate the performance of the EIM framework.

8 EXPERIMENTS

In this section, we experimentally evaluate the performance of the EIM framework on both real-world and synthetic evolving networks to investigate the following key issues. (1) Can the seed set selected by the EIM framework consistently outperform state-of-the-art methods in the EIM problem? (2) Does the particle learning method capture network growing speed and estimate node degrees well? (3) Does the running time of the EIM framework scale well in large scale networks? (4) What are the effects of the seed set size $K$ and the evolving time on the performance of the EIM framework? To answer the four questions, we will first introduce the evolving network datasets constructed in our experiments and then provide the detailed results.

ACM Transactions on Internet Technology, Vol. 20, No. 4, Article 40. Publication date: October 2020.
### Table 2. Statistics of Evolving Datasets

| Datasets | # of Nodes | # of Edges | Time Interval |
|----------|------------|------------|---------------|
| Co-author | 1.7M       | 12.6M      | A.D. 1801–2015 |
| Topic    | 34K        | 727K       | A.D. 1800–2016 |
| ML       | 1.51M      | 6.9M       | A.D. 1872–2017 |
| Bio      | 1.04M      | 1.82M      | A.D. 1992–2017 |
| SN       | 420 K      | 3.86M      | 25 periods    |

Fig. 2. Evolution of networks. (a) plots the node size of three networks over time. (b)–(e) plot the degrees of four nodes in each real network over time ($d_0$ denotes the initial degrees of the nodes.)

## 8.1 Evolving Network Datasets

Since existing widely used social network datasets lack complete information of the joining time of each node, we extract four real evolving networks from the Microsoft Academic Graph (MAG) [41]. Besides, we also generate a synthetic network following the Barabási-Albert evolving model [28]. The statistics of the five datasets are summarized in Table 2.

1. **Co-author**: From the author list of each paper, we extract a co-authorship evolving network that contains 1.7M nodes (authors) and 12.6M edges. If there is at least one paper co-authored by a pair of authors, then there is an edge between them. The joining time of each user is set as the publishing time of his first paper.

2. **Topic**: There are 127M papers in the MAG dataset, and we classify them into 34K topics with reliable ground-truth communities. The joining time of each topic is set as the publishing time of its earliest paper, and the edges correspond to the citations among topics.

3. **Machine Learning (ML)**: The evolving ML network consists of the papers belonging to the Machine Learning topic, which contains 1.51M nodes and 6.9M edges. The joining time of each node is set as the publishing time of its corresponding paper, and the edges are established based on the citations among papers.

4. **Bioinformatics (Bio)**: The evolving Bio network consists of the papers on the Bioinformatics topic, which contains 1.04M nodes and 1.82M edges. The construction method of Bio is similar to that of the ML network.

5. **Synthetic Network (SN)**: We also generate a synthetic network that includes 420K nodes and 3.86M edges based on the Barabási-Albert (BA) evolving model [28]. In the generation, with probability $\frac{1}{2}$, the new node randomly chooses neighbor nodes to establish edges; otherwise, the possibility of a node being chosen as neighbor is proportional to its current degree.

Figure 2 plots the growths of node sizes and degrees in the four real evolving networks. From Figure 2(a), we can find that Co-author, ML, and Bio all exhibit rapid growth of user size. From Figures 2(b)–2(e), we can see that the gap among the degrees of different users enlarges with network evolution. Especially, the node with the highest initial degree exhibits significant advantage on attracting new users in later years. Such phenomenon well justifies the BA evolving model that new users will preferentially connect to those with higher degrees in evolution.
8.2 Experimental Settings

Baselines. We compare the performance of the EIM framework with the following four baselines:

(1) IMM [8]: As a static IM framework, the objective of IMM in each round is selecting a set \( S_r \) of \( K \) seeds to maximize \( I(S_r, G') \). Then, recalling Section 6.2, the main idea of the IMM framework is first generating a sufficient number of RR-sets over deterministic copies of \( G' \) and then greedily selecting the seeds that can cover the most number of RR-sets.

(2) SKIM [15]: The SKIM is also a static IM framework with the objective being maximizing \( I(S_r, G') \). Similar to the IMM framework, a deterministic copy \( g \) of \( G' \) is first sampled, where each edge \( e \) is tagged as active with the activation probability \( w_e \) via it and as inactive with probability \( 1 - w_e \). Then, the SKIM iteratively conducts the BFS search processes over \( g \) starting from randomly chosen nodes. Each node \( u \) in \( G' \) is attached with an index set named \( \text{index}[u] \). When a BFS search process reaches a node \( u \), SKIM adds the starting node of the BFS search into \( \text{index}[u] \). By the time of the size of the index set \( \text{index}[u] \) first reaching a preset threshold, SKIM selects the node \( u \) as the first seed. Then, SKIM removes the nodes in \( \text{index}[u] \) from the index sets of other nodes and selects the following \( K - 1 \) seeds in a same manner.

(3) Highest Degree (HD): A heuristic algorithm that selects \( K \) seed users with the highest degrees in each trial.

(4) Earliest: A heuristic algorithm that selects \( K \) seed users who join the evolving network earliest in each trial.

Note that as the IMM and SKIM are two static IM frameworks, there is no degree and activation probability estimating methods in their original settings. Thus, when conducting IMM and SKIM, we set the degree of each node in the current network \( G' \) as their real current degrees and set the activation probabilities as the initial estimated value \( \overline{w}_0 \). We compare the influenced size of the seed sets returned by the EIM framework with IMM and SKIM to demonstrate the necessity of taking into account the influence diffused to future users and learning uncertain activating probabilities when conducting the IM campaign in evolving networks.

Parameter settings. For the four real network datasets, we set one year as the period for each trial and set the time for the first trial in the four datasets as (1) Co-author: 1965, (2) Topic: 1961, (3) ML: 1988, (4) Bio: 1993. Taking Co-author as the example, the initial network \( G^1 \) consists of the nodes joining network before 1965 and the relations among them. Then, the new nodes emerging during the first round are those who join network during 1966. For the synthetic network (SN), we conduct the first trial at the timestamp when network size is 2,500. Then, in SN, we set the size of new users joining the network during the \( r \)th round as \( 2,500 \cdot 2^r \).

Furthermore, the initial activation probabilities via edges are sampled from \( N(0.05, 0.008) \). Regarding the undirected citation patterns in Topic, ML, and Bio, we set the probability \( w_{AB} \) via the edge where \( B \) cites \( A \) as the Gaussian random walk and let \( w_{BA} \) always be 0. In bidirected networks Co-author and SN, the activation probabilities of two directions are both set as the Gaussian random walk, while the two activation probabilities are independent. The default number \( M \) of particles is set as 500, and effect of \( M \) will be shown later in Section 8.5. The default value of \( \varepsilon \) in the Sampling phase (Algorithm 4) is set as \( \varepsilon = 0.1 \), whose effect is further graphically reported in Section 8.7.

Environment. All the experiments are implemented in Python 2.7 and conducted on a computer running Ubuntu 16.04 LTS with 40 cores 2.30 GHz (Intel Xeon E5-2650) and 128 GB memory.

8.3 Effectiveness of EIM

We quantify the effectiveness by the number of influenced users, and report the comparison of the effectiveness between the EIM framework and four baselines in Figures 3 and 4.
Effects of evolving time. From Figure 3, we can observe that over the Co-author, ML, Bio, and SN, the EIM framework always outperforms the four baselines. The superiority of the EIM framework becomes more significant as time increases. Especially in the case of $K = 50$, Year = 2015 over Co-author, the influenced size of the EIM framework is almost 50% larger than that of the baselines. The superiority of the EIM framework owes to the continuous learning of network knowledge. Thus, with more accurate network knowledge, the EIM framework can return better seed set. This phenomenon justifies that the seed selection and the network knowledge learning can mutually enhance each other.

Over Topic, it can be observed that the influenced size of the EIM framework is smaller than that of HD in early years. The reasons behind are that: (1) The uncertainties of network knowledge degrade the performance of the EIM framework in early years. (2) Topic is the densest network where each new node (topic) averagely cites more than 20 existing nodes, and with higher probability, each new node cites the top 50 nodes with the highest degrees under the PA rule. However, even over the Topic network, the EIM framework still enjoys better performance than the four baselines in later years.
Effects of $K$. From Figure 4, we can find that the influenced sizes of IMM and SKIM grow smoothly with the increase of $K$, while the influenced size of Earliest presents much more fluctuations. The reason behind is that IMM and SKIM are efficient IM algorithms over static networks with rigorous performance guarantee, and their disadvantages to the EIM framework are brought by the inapplicability in evolving network. Meanwhile, Earliest is a heuristic, and the instability of its performance implies the heterogeneity of user attractiveness, especially among those joining network during early stage. In contrast, another heuristic HD achieves medium influenced size among the five algorithms, since new nodes preferentially connect with the nodes with high degrees.

8.4 Efficiency of EIM Framework

Now, we report the running time of the EIM framework. Figure 5 presents the running time of the EIM framework under $K = 5, 10, 20, 50$ against the network size in different years. As shown in Section 6.3, the complexity of the EIM framework in the $r$th trial is $O(M|V^r| + |E^r| + (K + l)(|V^r| + |E^r|) \log |V^r|/\varepsilon^2)$. Due to the high efficiency of the linear UCB and IMM frameworks,
the time cost of the EIM framework shown in Figure 5 linearly grows with the network size, and the EIM framework scales well even over networks of million scale. The running time of several classical IM algorithms over million-scale networks are: TIM (10^4 s) [14], TIM+ (10^3 s) [14], IMM (10^2 – 10^3 s) [8]. According to the experimental results in Reference [14], another two classical algorithms RIS and CELF++ cost 10^4 seconds over the network with 76K nodes. What we can also find from Figure 5 is that the increase of K from 10 to 50 only incurs slightly larger time costs, ensuring the scalability of the EIM framework in the cases where a large number of seeds need to be selected.

8.5 Performance of Learning Network Evolution

We evaluate the performance of learning network evolution from the accuracy of growing speed learning and the accuracy of degree prediction. Referring to Reference [17], the initial parameters are sampled from their possible ranges as: \( \beta \in [10^{-8}, 1] \), \( \theta \in [10^{-4}, 10] \), \( N \in [10^5, 10^8] \). On the performance of growing speed learning, we adopt the relative error, i.e., \(|(\sum_{i \in P} n_i(T'))/M - n(T')|/n(T')\) to measure the accuracy of learned network growing speed. Figure 6 plots the relative errors over Topic and SN with particle size being M = 500 and 1,000. We can find from Figure 6 that the accuracy of particle learning improves with the size of particles, since more particles bring higher resolution of initial parameters. The accuracy also increases over time, since particles with accurate parameters are gradually filtered out in the resampling phase in each trial.
For evaluating the performance of degree prediction, we also adopt the relative error, i.e., $\frac{|E(d_{n+1}^r) - d_{n+1}^r|}{d_{n+1}^r}$, where $E(d_{n+1}^r)$ denotes the estimated degree of node $u_n$ until $T^{r+1}$ and $d_{n+1}^r$ denotes the real degree of $u_n$ at $T^{r+1}$. We compute such relative error of each node at the end of each round and count the percentages of the relative errors belonging to the interval $<0.1, 0.1 - 0.3, 0.3 - 0.5, >0.5$, respectively. Figure 7 presents the percentages of relative errors in different years over Co-author, Topic, and SN. We can see from Figure 7 that, from the 12th round, the relative errors of estimated degrees of almost all nodes are smaller than 0.5. In the later stage, more than 90% of the relative errors are smaller than 0.1. Figures 6 and 7 demonstrate that the proposed growing speed learning and degree prediction methods in the EIM framework can well capture the network evolution.

### 8.6 Performance of Activation Probability Learning

For evaluating the performance of activation probability learning, we still adopt the relative error, i.e., $\frac{|w'_{ij,r} - w_{ij,r}|}{w_{ij,r}}$, where $w'_{ij,r}$ is the estimated activation probability from $u_i$ to $u_j$ in the $r$th trial and $w_{ij,r}$ is the real activation probability from $u_i$ to $u_j$. We compute such relative error of the estimated activation probability via each edge in each round and count the percentages of the relative errors belonging to the interval $<0.5, 0.5 - 1, >1$, respectively. Figure 8 presents the percentages of relative errors in different years over Co-author, ML, and SN. As the initial values of the activation probabilities are sampled from the distribution $\mathcal{N}(0.05, 0.008)$, most of the real activation probabilities are smaller than 0.1. From Figure 8, we can see that, in most cases, more than half of the relative errors of estimated activation probabilities are smaller than 1. Notably, since we adopt the UCB-based method to estimate the activation probabilities and intentionally increase the estimated activation probabilities via the edges with fewer triggered times, the relative errors are not very small. That is, we sacrifice the high accuracy of activation probability learning to balance the tradeoff between exploration and exploitation and maximize the influence diffusion size in a long run.
8.7 Effects of Parameter $\varepsilon$

Recalling Theorem 6.1, the approximation ratio and computational complexity are both the functions of parameter $\varepsilon$. Figure 9 shows the effects of $\varepsilon$ on the running time and influenced size over ML with $K = 20$. Since larger $\varepsilon$ means smaller number of required RR-sets in $\text{Evo-IMM}$, the running time of $\varepsilon = 0.5$ is much smaller than that of $\varepsilon = 0.1$. Although the increase of $\varepsilon$ causes the decrease of theoretical performance guarantee, $\text{Evo-IMM}$ experimentally achieves comparable influenced sizes in the two cases of $\varepsilon = 0.5$ and $\varepsilon = 0.1$.

9 CONCLUSION

This article investigates the evolving influence maximization in evolving networks where new users continuously join with influence diffusion. A bandit-based framework $\text{EIM}$ is proposed to simultaneously select seeds and learn network knowledges. In each trial, a particle learning-based method is first adopted to learn the network growing speed based on the preferential attachment rule. Then a UCB-based estimating method is designed to learn evolving activation probabilities among users. Based on the learned growing speed and activation probabilities, we propose an evolving IM algorithm $\text{Evo-IMM}$ to efficiently select the seed users for evolving IM. We show that the regret bound of the $\text{EIM}$ framework is sublinear to the number of trials. At last, the experiments on both real and synthetic evolving network datasets demonstrate that $\text{EIM}$ outperforms four baselines in solving the evolving influence maximization problem in evolving networks.

REFERENCES

[1] X. Wu, L. Fu, J. Meng, and X. Wang. 2019. Maximizing influence diffusion over evolving social networks. In Proceedings of the ACM SocialSense. 6–11.
[2] Q. Wu, Z. Li, H. Wang, W. Chen, and H. Wang. 2019. Factorization bandits for online influence maximization. In Proceedings of the ACM SIGKDD. 636–646.
[3] Y. Lin, W. Chen, and J. C. Lui. 2017. Boosting information spread: An algorithmic approach. In Proceedings of the IEEE ICDE. 883–894.
[4] K. Han, C. Xu, F. Gui, S. Tang, H. Huang, and J. Luo. 2018. Discount allocation for revenue maximization in online social networks. In Proceedings of the ACM MobiHoc. 121–130.
[5] G. Tong, W. Wu, S. Tang, and D. Du. 2016. Adaptive influence maximization in dynamic social networks. IEEE/ACM Trans. Netw. 25, 1 (2016), 112–125.
[6] J. Yuan and S. J. Tang. 2017. Adaptive discount allocation in social networks. In Proceedings of the ACM MobiHoc. 1–10.
[7] X. Wu, L. Fu, Y. Yao, X. Fu, X. Wang, and G. Chen. 2018. GLP: A novel framework for group-level location promotion in geo-social networks. IEEE/ACM Trans. Netw. 26, 6 (2018), 2870–2883.
[8] Y. Tang, Y. Shi, and X. Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In Proceedings of the ACM SIGMOD. 1539–1554.
[9] H. Zhuang, Y. Sun, J. Tang, J. Zhang, and X. Sun. 2013. Influence maximization in dynamic social networks. In Proceedings of the IEEE ICDM. 1313–1318.
[10] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart. 2015. Online influence maximization. In Proceedings of the ACM SIGKDD. 645–654.

[11] T. K. Shih, W. Gunaratne, A. Ochirbat, and H. M. Su. 2018. Grouping peers based on complementary degree and social relationship using genetic algorithm. ACM Trans. Internet Technol. 19, 1 (2018), 2.

[12] D. Kempe, J. Kleinberg, and É. Tardos. 2003. Maximizing the spread of influence through a social network. In Proceedings of the ACM SIGKDD. 137–146.

[13] H. T. Nguyen, T. P. Nguyen, T. N. Vu, and T. N. Dinh. 2017. Outward influence and cascade size estimation in billion-scale networks. In Proceedings of the ACM Sigmetrics. 1, 1 (2017), 1–30.

[14] Y. Tang, X. Xiao, and Y. Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In Proceedings of the ACM SIGMOD. 75–86.

[15] E. Cohen, D. Delling, T. Fajor, and R. F. Werneck. 2014. Sketch-based influence maximization and computation: Scaling up with guarantees. In Proceedings of the ACM CIKM. 629–638.

[16] Wechat. 2017. Retrieved from http://www.wechat.com/en/.

[17] A. C. Harvey. 1990. Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge University Press.

[18] H. Li, S. Bhowmick, J. Cui, and J. Ma. 2017. Time is what prevents everything from happening at once: Propagation time-conscious influence maximization. arXiv preprint arXiv:1705.10977 (2017).

[19] J. Iribarren and E. Moro. 2009. Impact of human activity patterns on the dynamics of information diffusion. Phys. Rev. Lett. 103, 3 (2009), 038702.

[20] N. T. Gayraud, E. Pitoura, and P. Tsaparas. 2015. Diffusion maximization in evolving social networks. In Proceedings of the ACM COSN. 125–135.

[21] Z. Wen, B. Kveton, M. Valko, and S. Vaswani. 2017. Online influence maximization under independent cascade model with semi-bandit feedback. In Proceedings of the NIPS. 3022–3032.

[22] Y. Yang, X. Mao, J. Pei, and X. He. 2016. Continuous influence maximization: What discounts should we offer to social network users? In Proceedings of the ACM SIGMOD. 727–741.

[23] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. 2012. Inferring networks of diffusion and influence. ACM Trans. Knowl. Discov. Data 5, 4 (2012), 21.

[24] Z. Shen, W. X. Wang, Y. Fan, Z. Di, and Y. C. Lai. 2014. Reconstructing propagation networks with natural diversity and identifying hidden sources. Nat. Commun. 5, 1 (2014), 1–10.

[25] S. Lamprier, S. Bourigault, and P. Gallinari. 2015. Extracting diffusion channels from real-world social data: A delay-agnostic learning of transmission probabilities. In Proceedings of the IEEE/ACM ASONAM. 178–185.

[26] M. Salathé, M. Kazandjieva, J. W. Lee, P. Levis, M. W. Feldman, and J. H. Jones. 2010. A high-resolution human contact network for infectious disease transmission. Proc. Nat. Acad. Sci. 107, 51 (2010), 22020–22025.

[27] C. Zeng, Q. Wang, S. Mokhtari, and T. Li. 2016. Online context-aware recommendation with time varying multi-armed bandit. In Proceedings of the ACM SIGKDD. 2025–2034.

[28] W. Chen, Y. Wang, and Y. Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In Proceedings of the ACM ICML. 151–159.

[29] W. Chen, C. Wang, and Y. Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In Proceedings of the ACM SIGKDD. 1029–1038.
[40] P. Rigollet and J. C. Hütter. 2019. High dimensional statistics: Lecture notes. Retrieved from http://www-math.mit.edu/rigollet/PDFs/RigNotes17.pdf.

[41] Microsoft Academic Graph. 2016. Retrieved from https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/.

[42] Xudong Wu, Luoyi Fu, Zixin Zhang, Huang Long, Jingfan Meng, Xinbing Wang, and Guihai Chen. 2019. Supplementary Material. Retrieved from http://www.cs.sjtu.edu.cn/fu-ly/paper/ElMfull_ACM_Tran_supp.pdf.

Received September 2019; revised April 2020; accepted June 2020