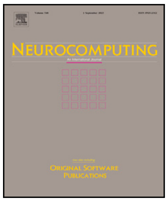
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Conformity-aware adoption maximization in competitive social networks

Yonggang Liu [a], Yikun Hu [a,*], Siyang Yu [a,b], Xu Zhou [a], Keqin Li [a,c]

[a] *College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China*
[b] *College of Information Technology and Management, Hunan University of Finance and Economics, Changsha 410000, China*
[c] *Department of Computer Science, State University of New York, New Paltz, NY 12561, USA*

ARTICLE INFO

Communicated by N. Zeng

Keywords:
Adoption maximization
Conformity
Competitive influence
Reverse influence sampling

ABSTRACT

Influence maximization (IM) problem is an extensively studied problem in social networks. It aims to find a small set of users in the social network to initiate the diffusion process and maximize the expected influence spread. Existing works on conformity-aware IM focus on the interaction between influence and conformity in a single-influence setting and ignore the role of conformity in a competitive and multiple-influence setting. This paper proposes a conformity-aware independent cascade (C-IC) model that considers the competition among multiple influences as well as the role of conformity in a user's decision-making. It is proved that the adoption of an influence under the C-IC model is monotone and submodular. Meanwhile, we formulate two adoption maximization (AM) problems, O-AM and S-AM, which are both NP-hard. Because estimating the adoption through diffusion simulations is very time-consuming, we propose a reverse adoption estimation (RAE) method based on a reverse multiple influence sampling (RMIS) technology for the C-IC model and integrate it into the D-SSA-fix (Nguyenet al., 2018) framework, DSSA for short, to compute a solution with approximation guarantee. To further boost the performance, we present a fast one-hop adoption estimation (OAE) method and develop a heuristic algorithm based on OAE, called GOAE. Extensive experiments on eight real-world social networks show that the C-IC model is superior to a non-conformity diffusion model and that RAE+DSSA and GOAE are efficient and effective. In most cases, GOAE finds comparable solutions to RAE+DSSA and CELF with less time and memory overhead. GOAE is five to six orders of magnitude faster than CELF and RAE+DSSA is up to three orders of magnitude faster than CELF on NetHEPT. GOAE runs up to four to five orders of magnitude faster than RAE+DSSA with at most two orders of magnitude less memory usage. GOAE is more scalable than RAE+DSSA in terms of the number of seeds and the size of the social network.

1. Introduction

In the past decades, much work on social networks has emerged in sociology, economics, psychology, and other disciplines [1,2]. As information technology advances [3–5], with the emergence of growing large-scale datasets from online social networks, social network analysis draws a great deal of interest from researchers but faces many new challenges.

Challenge. With the development of communication technology and the Internet, communication is more convenient and online contents are easier to obtain. People can easily access more diverse information from social networks. For example, online social platforms like TikTok and Xiaohongshu in China are growing in popularity, and users enjoy sharing their experiences with their followers. These experiences often involve multiple (one or more) similar products, such as newly released mobile phones or some latest movies. As a result, the followers may receive information on different products from a user or several users These people who receive diverse influences often face a problem that how to make a choice. And the online social platforms also want to know how to select the most influential users in their platforms. This poses a challenge for modeling the diffusion and adoption of multiple influences in competitive social networks. However, the existing diffusion models are not suitable for this context. Most of the existing diffusion models are designed for single-influence settings. However, in real life, there are numerous competitive products in the social network and people may even recommend multiple similar products simultaneously based on their experience or information she has learned. In addition, even for the existing competitive multiple-influence diffusion models, they often use a "first come, first served" strategy or a fair strategy to choose a product, which may lead to biased adoption estimation or the selection of users who are not the most influential.

[*] Corresponding author.
E-mail addresses: liuyonggang@hnu.edu.cn (Y. Liu), yikunhu@hnu.edu.cn (Y. Hu), yusiyang@hufe.edu.cn (S. Yu), zhxu@hnu.edu.cn (X. Zhou), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.neucom.2023.127224
Received 6 September 2023; Received in revised form 26 November 2023; Accepted 30 December 2023
Available online 3 January 2024

To overcome this challenge, we propose a conformity-aware independent cascade (C-IC) model that considers the competition among multiple influences and the role of conformity in a user's decision-making. The C-IC model emphasizes the role of the frequency of receiving an influence in a user's decision-making and leverages the property of conformity to build a bridge between the frequency of receiving an influence to a user's adoption probability. [6–8] conducted similar field experiments in the USA, the UK, and Sweden to study the relationship between conformity and the number of influencers, respectively. They find as the number of influencers increased. Similar phenomena also appear online, where more people have more influence. [9] reported users on Facebook were more likely to like content if they saw three (compared to one) people had liked it. [10] reported people were less likely to believe a news if several others commented it was untrue when they reading on social media. Therefore, we assume that the adoption probability of an influence increases with the frequency of receiving an influence.

On the other hand, viral marketing aims to promote and popularize products and innovations by word of mouth to new users who often lack sufficient knowledge of the products and innovations. New users' decisions are usually influenced by informational conformity [2]. Informational conformity stems from the mentality that when individuals lack sufficient knowledge and experience, they tend to trust the wisdom of the group and align themselves with it. For example, on social platforms, individuals tend to believe the messages that are frequently retweeted, even if they do not know the truth. Moreover, individuals are more likely to adopt the products frequently recommended by friends in their social networks. These phenomena also verify that the C-IC model is reasonable.

Based on the C-IC model, we formulate two adoption maximization problems, named O-AM and S-AM, which are significant in applications such as viral marketing and innovation promotion. The adoption maximization problem aims to find a small set of users to start the propagation process and maximize the adoption under the C-IC model. In the AM problem, adoption serves as the criterion for evaluating the quality of a solution instead of the influence spread, since an influenced user only adopts one influence from the influences she received under the C-IC model.

Contributions. In this paper, we present the C-IC model in the context of multiple competitive influences, which portrays the diffusion process of multiple competitive influences and highlights the role of conformity in users' decision-making. Besides, we formulate two adoption maximization problems, named O-AM and S-AM, under the C-IC model and develop two methods, RAE+DSSA and GOAE, to address the S-AM problem. To summarize, our contributions are as follows.

- We propose the C-IC model for competitive social networks and demonstrate that the expectation of adoption under the C-IC model is non-negative, monotone, and submodular. Then we present two AM problems, O-AM and S-AM, and prove they are NP-hard.
- We present the RAE method based on the RMIS technology for the C-IC model to estimate the adoption and integrate it into the DSSA [11] framework for the S-AM problem. Then, we demonstrate the mathematical basis of the RAE method.
- We provide a GOAE algorithm based on a fast one-hop adoption estimation (OAE) which further speeds up the adoption estimation. The GOAE algorithm is more scalable than RAE+DSSA in terms of running time and memory usage, particularly in the context of large seed numbers and large-scale networks.
- We conducted experiments on eight real-world network datasets. We evaluate the C-IC model by compared it with a non-conformity diffusion model. The experiment results show that the C-IC model is superior to a non-conformity diffusion model and more conducive to obtaining superior seed sets. We compare RAE+DSSA, GOAE, and CELF. These experiments suggest that RAE+DSSA and GOAE are efficient and effective. GOAE finds comparable solutions to RAE+DSSA and CELF with less time and memory usage. The experiments with the large k setting suggest GOAE scales well for large k values on large networks.

We organize the rest of the paper as follows. We introduce the related work in Section 2 and the preliminaries of this work in Section 3. We propose the C-IC model and formulate the S-AM problem and the O-AM problem in Section 4. We establish the theoretical basis of the RAE method and introduce the RMIS technology in Section 5. We present the GOAE algorithm based on the OAE method in Section 6. We illustrate the experiments on eight real-world network datasets and analyze the results of the experiments in Section 7. We conclude this paper in Section 8. For ease of reading, all the proofs are given in Appendix.

2. Related work

Kempe et al. [12] formally proposed the influence maximization (IM) problem, and then proved that it is NP-hard. Additionally, they proved the influence spread functions under the independent cascade (IC) model and the linear threshold (LT) model are both non-negative, monotone, and submodular, and based on the fact they proposed a greedy algorithm to find a solution providing a $(1-1/e-\varepsilon)$-approximation. Leskovec et al. [13] proposed the CELF (Cost-Effective Lazy Forward) algorithm to accelerate seed search by avoiding estimating the influence spreads of unnecessary candidate solutions benefitting from the submodularity of the influence spread function. Some existing algorithms for IM waive approximation guarantees for improving practical efficiency. Chen et al. [14] proposed DegreeDiscount which is significantly superior to the degree and centrality-based heuristics and runs faster than the greedy algorithms in [12,13] by many orders of magnitude. Chen et al. [15] showed that computing influence spread in the IC model is #P-hard. To improve the performance of computing influence spread, they designed two heuristic algorithms, called MIA and PMIA, which use local arborescence structures of each node to approximate the influence spread. Jiang et al. [16] proposed a simulated annealing-based approach for the IM problem to replace the time-consuming greedy algorithm. To further improve the efficiency, they use EDV (expected diffusion value) instead of estimating the influence spread through influence diffusion simulations. Jung et al. [17] proposed IRIE which derives a system of linear equations whose solution can be computed fast by an iterative method. Then the computed values are used as estimations of the influence spread.

More recently, a wealth of extensions on IM have emerged. Kazemzadeh et al. [18] proposed the IMBC (Influence Maximization Based on Community structure) algorithm which exploits optimal pruning and a minimum of dominating nodes to improve efficiency and modulates the scores of nodes with a high Rich-Club coefficient to optimize the selection of seed set. Yang et al. [19] proposed a Continuous Influence Maximization problem based on an assumption that the purchase probability curve with respect to discount for each user is known. They investigated what discounts should be offered to users to maximize the adoption of a product. Ohsaka et al. [20] proposed an algorithm to coarsen an original influence network into a node-weighted influence network which is much smaller and can approximate the diffusion properties of the original influence network. The coarsened influence network is used to speed up the estimation of influence spread and the algorithms for IM. Zhao et al. [21] transformed identifying the most influential nodes into a classification problem and proposed an InfGCN model based on Graph Convolutional Networks (GCN), which considers the roles of both network structures and node features in identifying the importance of nodes. Kou et al. [22] transformed identifying influential nodes into a regression problem and proposed a deep learning model based on the graph multi-head attention mechanism and the dense connection to identify the most influential nodes.

2.1. Competitive IM and adoption maximization.

Bharathi et al. [23] are among the first who study the competitive IM problem with multiple competing innovations. Bhagat et al. [24] presented an LT-C model and studied the adoption maximization based on it. They distinguish between adopting and influencing and take users' attitude into account based on her experience with products. They showed that the adoption maximization problem is NP-hard and the expected number of product adoptions is monotone and submodular under the LT-C model; Valera and Gomez-Rodriguez [25] proposed a continuous-time probabilistic model, based on temporal point processes, for the adoption and frequency of use of competing products, which captures several intuitive key factors, i.e. social influence, recency, and competition. Li et al. [26] proposed a game theory-based framework for the competitive IM problem which jettisons unrealistic assumptions that a new competitor is aware of a rival's strategy. Zhu et al. [27] presented Competitive Independent Cascade model in which users including seeds are able to spread competitive influences at the same time and investigated the Minimum Cost Seed Set problem based on their model. Recently, Hong et al. [28] presented a competitive reverse influence estimation-based greedy (CRIEG) algorithm with bounded approximation guarantees, which significantly improves efficiency under the competitive IC model.

2.2. RIS-based algorithms.

Time efficiency becomes a primary challenge for the IM problem due to the increasing size of social networks. Diffusion simulation-based greedy algorithms are extremely time-consuming and not scalable, while other heuristic algorithms lack approximation guarantees. Borgs et al. [29] made a theoretical breakthrough and proposed a novel $O(kl^2(m+n)\varepsilon^{-3}log^2n)$ time algorithm based on a drastically different method which is known as reverse influence sampling (RIS) for the IM problem under the IC model. Tang et al. [30] further reduced the running time to $O((k+l)(m+n)logn/\varepsilon^2)$ and proposed two algorithms, TIM and TIM+, for the IM problem under the triggering model which is a general diffusion model including the IC model and the LT model. Then, they [31] proposed a further improved algorithm, IMM, for the IM problem, which can support any diffusion model for which a certain sampling procedure is well-defined. Nguyen et al. [32] adopted a Stop-and-Stare strategy and proposed two algorithms, SSA and D-SSA, which perform better than IMM in terms of empirical efficiency. Huang et al. [33] uncovered some errors in proofs for the approximation factors and the sampling efficiency of SSA and D-SSA, and then provided an SSA-fix algorithm. Nguyen et al. [11] provided an D-SSA-fix algorithm and affirmed the sampling efficiency of D-SSA-fix. Wang et al. [34] proposed a bottom-k sketch based RIS framework (BKRIS), which brings the order of samples into the RIS framework, to accelerate the RIS framework and reduce memory consumption. Guo et al. [35] presented a framework for generating reverse reachable (RR) sets, called SUBSIM, and developed the SKIP algorithm for the sorted subset sampling problem. Then they presented the HIST algorithm to enhance the scalability in high influence networks. Zhu et al. [36] proposed a 2-hop+ sampling method for fast and accurate estimation of influence spread under the IC model, which reduces the sample number by generating only samples including at least one 2-hop live path. Then, they exploit a SkipEdge technique to further improve the sampling efficiency of their method. In addition, they presented the generalized stopping rule algorithm to obtain an (ε,δ)-estimation of the mean of random variables with fewer samples needed.

2.3. Conformity and conformity-aware social influence analysis.

Conformity is a fundamental and well-studied concept in social psychology. Extensive work in social psychology [2,37,38] has shown the importance of conformity and studied the relationship between conformity and the number of influencers [6–10,39]. Milgram et al. [6] conducted a field experiment in New York. They asked 1, 3, 5, 10, and 15 people, called influencers, to stop and look upwards on a busy sidewalk. They found the proportion of passers-by who are influenced and look upwards increases with the number of influencers. Coultas and Eriksson [7] and Gallup et al. [8] replicated the experiment in the UK and Sweden. They found as the number of influencers increased, the influence showed a similar linear pattern. Similar phenomena also appear online, where more people have more influence. Egebark and Ekstrom [9] reported users on Facebook were more likely to like content if they saw three (compared to one) people had liked it. Colliander [10] reported people were less likely to believe a piece of news if several others commented it was untrue when they reading on social media. Based on these observations, we model conformity in the C-IC model as a social influence that increases with the times a user receives it. Li et al. [40] studied conformity as an individual's inclination to be influenced by others and they computed conformity indices of each individual by using individuals' relationships with positive or negative signs. Zhang et al. [41] studied how the conformity tendency changes with users' role defined by her structural properties and proposed a probabilistic graphical model for modeling the role-aware conformity influence. Tang et al. [42] studied the role of conformity in changing individuals' online behavior and formalized the effects of social conformity into a probabilistic model. The three works study conformity for computing individuals' traits or predicting individuals' actions, which is different from the purpose of this paper. Li et al. [43] proposed a conformity-aware cascade (C^2) model that exploits the influence probabilities computed with conformity in [40] for estimating influence spreads in the context of the IM problem in signed social networks. Li et al. [44] proposed a group-based algorithm for the IM problem under the conformity-aware diffusion model based on user profiles and group profiling. Recently, Li et al. [45] proposed a conformity-aware Hawkes process-based framework to characterize online information diffusion and used a semi-parametric inference approach to learn their model. The three works study online information diffusion or the IM problem in a different setting from this work. And they cannot to be used for competitive and multiple-influence diffusion environments which is the context of this work.

Table 1
Frequently used notations.

Notation	Definition
$G(V,E,p)$	an influence graph
$N_{in}(u)$	the in-neighbor set of u
$N_{out}(u)$	the out-neighbor set of u
$N_A(u)$	the set of in-neighbors activated u
S (S_i)	a seed set (the seed set of influence I_i)
S_c	the seed set of all competitive influences
$\mathcal{S}_c$	the set of seed sets of all competitive influences
$\mathcal{I}$	the influence set
$\mathcal{I}(u)$	the set of influences received by u
$h(u,I)$	the probability u adopts an influence I
$\sigma(S)$	the influence spread of a seed set S
$f(S)$	the adoption of a seed set S

3. Preliminaries

A social network is modeled as an influence graph $G=(V,E,p)$, where V is the set of nodes, E is the set of directed edges, with $n=|V|$ and $m=|E|$ and $p: E \rightarrow (0,1]$ is an influence probability function. $N_{in}(u)$ and $N_{out}(u)$ represent the in-neighbor (incoming neighbor) set and the out-neighbor (outgoing neighbor) set of a node u, respectively.

In this section, we review the IC model and the RIS technology under the IC model. Table 1 lists notations used frequently.

3.1. Independent cascade model

The independent cascade model is one of the most widely used diffusion models in the IM problem. In the IC model, each node takes one of two states, active or inactive. Inactive nodes can become active, but not vice versa. Given an influence graph $G = (V, E, p)$ and an initially activated seed set $S \subseteq V$. The diffusion proceeds in discrete steps according to the following randomized rules. When a node becomes active in Step t, it acquires only one chance to activate each of its inactive out-neighbors randomly with the influence probability between them. If any inactive out-neighbor is activated successfully, then it will become active in Step $t + 1$. The diffusion process ends when no more nodes are activated. Note that if an inactive node is activated by any node, it cannot be influenced by other in-neighbors. This is different from the C-IC model.

The influence spread $\sigma(S)$ is defined as the expected number of active nodes when the diffusion ends. Unfortunately, computing $\sigma(S)$ under the IC model is #P-hard [15]. The diffusion simulation-based method has been used for approximately computing $\sigma(S)$, which repeatedly performs diffusion simulations and takes the mean of the number of active nodes as $\sigma(S)$. In practice, ten thousand simulations are sufficient to estimate $\sigma(S)$ [12].

3.2. Influence maximization problem

The IM problem aims to find an optimal seed set in the social network to maximize the number of activated nodes under a diffusion model. It is defined formally as follows.

Definition 3.1 (*Influence Maximization Problem [12]*). Given an influence graph $G = (V, E, p)$ and an integer k, the influence maximization problem requires finding a seed set $S \subseteq V$ of size k that maximizes the influence spread $\sigma(S)$.

It is proved that computing the optimal seed set under the IC model is NP-hard [12]. Fortunately, the influence spread under the IC model is non-negative, monotone, and submodular. Therefore, the greedy algorithm can be used to look for a $(1 - 1/e - \varepsilon)$-approximate optimal solution [12]. The greedy algorithm starts with $S = \emptyset$. Then it chooses a node providing the largest marginal gain and adds the node into S iteratively until S includes k nodes.

However, the greedy algorithm for IM suffers from low efficiency because of two disadvantages. Firstly, computing the influence spread of a candidate seed set needs to perform large numbers of diffusion simulations. Secondly, it executes global searches in the influence graph for each new seed. In other words, to find a new seed, it estimates the influence spreads for all candidate seed sets. It runs $\sum_{i=0}^{k-1} (n - i) \cdot l$ simulations for finding a seed set of size k, where l is the number of simulations for computing the influence spread of each seed set. The greedy algorithm is computationally prohibitive for the IM problem, especially in large networks.

3.3. RIS under the IC model

Borgs et al. [29] developed the reverse influence sampling (RIS) technology which estimates $\sigma(S)$ by generating a set $\mathcal{R}$ of random reverse reachable (RR) sets. Each RIS process generates a random RR set R in the reverse influence graph G^T which is the transpose graph of the influence graph G. Note that the influence probability of each reversed edge in G^T is the same as the probability of the original edge in G. Given G^T, the RIS is like a random breadth-first traversal (BFS) from a randomly selected node u. A RIS proceeds as the following procedure: (1) Select a node u uniformly at random from G^T. (2) Visit u and start a random BFS from u in G^T. In the random BFS, each visited node gets one chance to randomly visit each of its out-neighbors with the influence probability between it and its out-neighbor. (3) The random BFS ends if no new nodes are visited. (4) Finally, return the set of the visited nodes as the random RR set R. Intuitively, u can reach the nodes in R in the RIS. Therefore, the nodes in R can influence u.

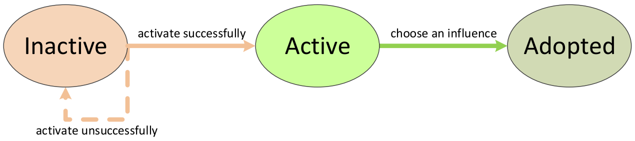


Fig. 1. Node states under the C-IC model.

Given a seed set S of an influence I, if any seed in S exists in R, i.e., S intersects R, u will be influenced by I. Given a set $\mathcal{R}$ of random RR sets, the more random RR sets S intersects, the more influential S is. Based on the idea, [29] uses a greedy algorithm to select the seed set that intersects the most random RR sets in $\mathcal{R}$. In addition, the RIS-based algorithms [29–36] for the IM problem investigate how to generate random RR sets with lower computational overheads to guarantee the quality and reliability of the seed set.

4. Problem definition

In real life, users may receive information on different products or innovations from their friends and form their own opinions and attitudes based on this information. In addition, research work in the field of social psychology shows people usually tend to conform with their friends. Therefore, we develop a conformity-aware independent cascade model that models the propagation and adoption of multiple competitive influences. Then, we define two adoption maximization problems under the C-IC model.

4.1. Conformity-aware independent cascade model

In the C-IC model, there are multiple influences propagating in the social network. Nodes including seeds can propagate multiple influences simultaneously. The seed sets of different influences may overlap and a seed may serve multiple influences. Furthermore, the C-IC model consists of two stages, activation and adoption. Nodes may receive multiple influences and spread all of them, but they can only adopt one. The adoption process is subject to conformity in the C-IC model which emphasizes the role of conformity in the adoption stage from an audience's perspective.

To facilitate the description of the C-IC model, some symbols are defined first. Denote by $\mathcal{I}$ the overall set of influences propagating in the social network G and denote by $\mathcal{I}(u)$ the set of influences received by u. $S = \bigcup_{i=1}^{|\mathcal{I}|} S_i$ is the overall seed set of all the influences, where S_i is the seed set of I_i.

In the C-IC model, each node has three states, inactive, active, and adopted as shown in Fig. 1. Each node keeps inactive before it is exposed to any influence. Only inactive nodes can be activated. Nodes will become active after they are activated and receive some influences. Each active node attempts to activate its inactive out-neighbors. Then they become adopted after they adopt one influence from the influences received. The adopted nodes remain adopted until the diffusion process ends. Nodes can only change their states in two ways, becoming active from inactive and becoming adopted from active.

Given an influence graph G and a seed set S, the diffusion proceeds in discrete time steps according to the following rules. To start the propagation process, the influences activate their seeds, while the rest nodes remain inactive. In Step t, each active node gets one chance to randomly activate each of its inactive out-neighbors with the influence probability between it and its out-neighbor. If a node u activates its out-neighbor v successfully, it propagates all the influences in its influence set $\mathcal{I}(u)$ to v. After attempting to activate all the out-neighbors, each active node randomly adopts one influence from its influence

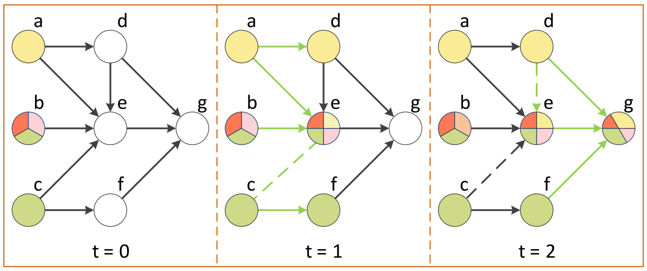


Fig. 2. An example of the influence diffusion process under the C-IC model.

set according to the conformity-aware adoption probability introduced in Section 4.2. Then the active nodes turn adopted and they cannot activate any inactive out-neighbor in subsequent steps. Note that if an inactive node is activated by its in-neighbors, it will remain inactive (activatable) until the next time step to give other active in-neighbors opportunities to activate it. As a result, a node may be activated by multiple in-neighbors in the same time step and may receive multiple influences from those in-neighbors. If an inactive node v is activated by an in-neighbor, v will receive all the influences in the set of the in-neighbor. Then we get $\mathcal{I}(v) = \bigcup_{u\in N_A(v)} \mathcal{I}(u)$, where $N_A(v)$ consists of all the in-neighbors that have activated v. In Step $t+1$, the nodes activated in Step t become active. Then they repeat the behaviors of active nodes in Step t. Finally, the diffusion process ends when no more nodes are activated.

Example. Fig. 2 illustrates the diffusion process of four influences under the C-IC model in a social network composed of seven nodes and nine directed edges. I_y, I_b, I_p, and I_g are represented in yellow, brown, pink, and green, respectively. The colors of a node are used to express how it is activated by corresponding influences. A green edge represents that an activation attempt occurs between the end nodes of the edge in the current time step. The dashed edges indicate that activation attempts fail or no activation occurs.

In Step 0, to start the influence diffusion, a, b, and c are activated and their influence set consists of the influences activating them. For example, $\mathcal{I}(b)$ becomes $\{I_b, I_p, I_g\}$, since node b has been activated by the three influences.

In Step 1, a, b, and c become active, and then attempt to activate their inactive out-neighbor along the green edges. Note that a and b activate e in the same time step, while c fails to activate e. Therefore, $\mathcal{I}(e)$ becomes $\{I_y, I_b, I_p, I_g\}$. After all activation attempts finish, e randomly adopts one influence from the received influences based on their adoption probabilities and becomes adopted.

In Step 2, d, e, and f become active and they successfully activate g. Note that g is activated by I_y and I_g twice and by I_b and I_p once. The sector area is used to indicate the ratio of activation times of different influences. For example, the green sector covers one-third of the area of g because the activation time of I_g is one-third of the total activation time. Then, d, e, and f adopt one influence from their influence sets and become adopted.

In Step 3, g becomes active, but no nodes can be activated. Then it randomly adopts one influence and becomes adopted. Finally, the diffusion process ends since no nodes are activated in this step.

4.2. Conformity-aware adoption probability function

Conformity is a kind of social influence that persons tend to fit in with their groups in social interactions. Conformity plays an important role when users make a choice in a competitive social network, especially when they lack sufficient knowledge about the products and innovations. In the adoption stage of the C-IC model, the conformity-aware adoption probability is defined as a function depending on the number of times the influences are received.

Denote by $h(u, I_i)$ the probability function that a node u adopts an influence I_i. We have $0 \le h(u, I_i) \le 1$ and $\sum_{I_i\in\mathcal{I}} h(u, I_i) = 1$ for any influence I_i and any activated node u. Specially, if an influence I_i is not exposed to u, $h(u, I_i) = 0$. And if only one influence I_i is exposed to u, $h(u, I_i) = 1$.

The conformity-aware adoption probability is modeled based on the following two observations of common phenomena in information diffusion process.

Observation one: Literature [6–10] in social psychology report that the conformity influence increases with the number of influencers and follows a similar linear pattern. Besides, to ensure a good decision is made, an individual usually supports the choice that the majority of people make and the probability of supporting a choice increases with the population supporting the same choice. We assume that the adoption probability of an influence increases with the number of times the influence is received.

Observation two: When a person learns about a product for the first time, it makes a deep impression. Nevertheless, the impact will wane as the number of times the same information is received increases. In other words, the marginal gain of the adoption probability of an influence decreases as the number of times the influence is received increases.

4.2.1. Linear adoption probability

Based on the two observations above, the conformity-aware adoption probability function is defined as

$$h(u, I_i) = H(N_A(u, I_i)) = \frac{|N_A(u, I_i)|}{\sum_{I_j\in\mathcal{I}} |N_A(u, I_j)|},$$

where set $N_A(u, I_j) \in N_{in}(u)$ consists of all the neighbors that propagate influence I_j to node u.

We discuss the properties of the adoption probability function for an influence I based on an assumption that seeds of competitive influences remain unchanged. We have the following conclusion.

Corollary 4.1. *The adoption probability function $h(u, I_i) = H(N_A(u, I_i))$ is a function from $N_{in}(u)^2$ to R, where $N_{in}(u)^2$ is the power set of $N_{in}(u)$ and R is the set of real numbers. $h(u, I_i)$ is non-negative, monotone, and submodular.*

Corollary 4.2. *For any set $X \subseteq Y \subseteq N_{in}(u)$ and any set $W \subseteq N_{in}(u) - Y$, we have*

$$H(X \cup W) - H(X) \ge H(Y \cup W) - H(Y). \tag{1}$$

4.3. Adoption under the C-IC model

The adoption $f(S_i)$ under the C-IC model is the expected number of nodes that adopt an influence I_i by using its seed set S_i. Additionally, the overall adoption $\mathcal{F}(S) = \sum_{i=1}^{|\mathcal{I}|} f(S_i)$ is the sum of adoptions of all the influences.

Theorem 4.3. *Given the seed set $S = \bigcup_{i=1}^{|\mathcal{I}|} S_i$, the overall adoption $\mathcal{F}(S)$ is non-negative, monotone, and submodular. Therefore, $\mathcal{F}(S)$ will reach the maximum when the seed sets of all the influences do not overlap and $|S| = \sum_{i=1}^{|\mathcal{I}|} |S_i|$ is the maximum.*

In the rest of this section, we focus on analyzing the adoption of a specific influence.

Theorem 4.4. *Given the influence graph $G = (V, E, p)$ and the set $S_c = \{S_j | j \ne i, 1 \le j \le |\mathcal{I}|\}$, the adoption $f(S_i)$ of the influence I_i under the C-IC model is non-negative, monotone, and submodular.*

We illustrate the complexity of computing the two adoption functions below.

Theorem 4.5. *Computing the overall adoption $\mathcal{F}(S) = \sum_{i=1}^{|\mathcal{I}|} f_{(S_i)}$ under the C-IC model is #P-hard.*

Theorem 4.6. *Given the set of competitive seed sets $S_c = \{S_j | j \neq i, 1 \leq j \leq |\mathcal{I}|\}$, Computing the adoption $f(S_i)$ of I_i on S_i under the C-IC model is #P-hard.*

4.4. Problem definition

Given an influence graph G, the number of seeds k, and the adoption probability function $h(u, I_i)$, we define the overall adoption maximization (O-AM) and the single adoption maximization (S-AM) under the C-IC model.

Definition 4.1 (*Overall adoption maximization (O-AM)*)**.** O-AM aims to find out a seed set $S = \bigcup_{i=1}^{|\mathcal{I}|} S_i$ that maximizes the overall adoption $\mathcal{F}(S) = \sum_{i=1}^{|\mathcal{I}|} f_{(S_i)}$ under the C-IC model.

Since $\mathcal{F}(S)$ will reach the maximum when the seed sets of all the influences do not overlap and $|S| = \sum_{i=1}^{|\mathcal{I}|} |S_i|$ is the maximum. Moreover, $\mathcal{F}(S)$ under the C-IC model is equal to $\sigma(S)$ under the IC model. Therefore, the O-AM problem under the C-IC model is equivalent to the IM problem under the IC model, which asks for a seed set S to maximize $\sigma(S)$. Furthermore, the O-AM problem under the C-IC model is NP-hard, since the IM problem under the IC model is NP-hard.

Definition 4.2 (*Single adoption maximization for (S-AM)*)**.**. Given set $\{S_j | j \neq i, 1 \leq j \leq |\mathcal{I}|\}$ of all competitive influences, S-AM for I_i requires finding out a seed set S_i of size k that maximizes the adoption $f(S_i)$ under the C-IC model.

The S-AM problem under the C-IC model is NP-hard, since when $|\mathcal{I}| = 1$, S-AM is equivalent to the IM problem under the IC model. Therefore, S-AM is not easier than IM which is NP-hard.

5. Reverse adoption estimation

Computing the adoption $f(S_i)$ of an influence under the C-IC model is #P-hard and the S-AM problem is NP-hard. The diffusion simulation-based method can be used to estimate $f(S_i)$. Benefitting from the nonnegativity, monotonicity, and submodularity of $f(S_i)$, the greedy algorithm can be used to tackle the S-AM problem and provides a $(1 - 1/e - \epsilon)$-approximate optimal solution. The greedy algorithm runs $\sum_{i=0}^{k-1} (n - i)$ adoption estimations to obtain k seeds for the S-AM problem. However, the running time of the greedy algorithm is prohibited, especially for large scale networks.

To overcome the challenges, we propose a reverse adoption estimation (RAE) method based on a reverse multiple influence sampling (RMIS) technology for the C-IC model for the S-AM problem.

We first demonstrate the mathematical basis of the RAE method in Section 5.1. Then we describe the RMIS process and explain how to compute the adoption probability under the C-IC model in Section 5.2. After that, we illustrate how to select seeds by updating the marginal adoption gains of candidate seeds in Section 5.3. Finally, the two technologies are integrated into the DSSA framework [11] in Section 5.4.

5.1. Mathematical basis of the RAE method

The RAE technology is proposed to bridge the gap between estimating adoption and the C-IC model, which avoids the inefficiency of estimating the adoption by repeatedly simulating the influence diffusion processes. It is based on the idea that the expected adoption $f(S_i)$ of the influence I_i under the C-IC model can be estimated by estimating the expected adoption probability $E[h(u, I_i)]$.

Table 2
Notations in Algorithm 1.

Notation	Definition
u	the sampling source selected randomly
n_r	a structure representing an RR node
$n_r.id$	a number representing an RR node
$n_r.level$	the distance from u to n_r
$n_r.N_u$	the neighbor set of u in the paths from u to n_r
g_r	a structure corresponding to an RR graph
$g_r.N_r$	the set consisting of all RR nodes
$g_r.d_c$	the distance from competitive seeds to u
$g_r.N_r(v)$	the RR node structure in $g_r.N_r$ whose id equals v
$N_u(I)$	the set of u's neighbors spreading I to u

Algorithm 1 RMIS

```
Input: The reverse influence graph $G^T$, a randomly selected node $u$, and the
    competitive seed set $S_c$
Output: A structure corresponding to the random RR graph, namely $g_r$
struct{$id$; $level$; $N_u$; }$n_r$;
struct{$N_r$; $d_c$; $t_c$; }$g_r$;
$g_r.d_c \leftarrow \infty$ and $g_r.t_c \leftarrow 0$;
$level \leftarrow 0$ and $seeds_c \leftarrow \emptyset$;
$n_r.id \leftarrow u$, $n_r.level \leftarrow level$, and $n_r.N_u \leftarrow \emptyset$;
$g_r.N_r \leftarrow g_r.N_r \bigcup \{n_r\}$;
if $u \in S_c$ then
    $g_r.d_c \leftarrow level$ and $g_r.t_c \leftarrow |\mathcal{I}(u)|$;
    Return $g_r$;
$seen \leftarrow \{u\}$, $active \leftarrow \emptyset$, and $next \leftarrow \{u\}$;
while $next \neq \emptyset$ && $seeds_c = \emptyset$ do
    $active \leftarrow next$ and $next \leftarrow \emptyset$;
    $level \leftarrow level + 1$;
    for each $v \in active$ do
        for each $w \in N_{out}(v)$ do
            if $w \notin seen$ then
                if $v$ activates $w$ then
                    $next \leftarrow next \bigcup \{w\}$ and $seen \leftarrow seen \bigcup \{w\}$;
                    $n_r.id \leftarrow w$ and $n_r.level \leftarrow level$;
                    if $level = 1$ then
                        $n_r.N_u \leftarrow \{w\}$;
                    else
                        $n_r.N_u \leftarrow g_r.N_r(v).N_u$;
                    $g_r.N_r \leftarrow g_r.N_r \bigcup \{n_r\}$;
                    if $w \in S_c$ then
                        $g_r.d_c \leftarrow level$ and $seeds_c \leftarrow seeds_c \bigcup \{w\}$;
            else
                if $w \in next$ && $v$ activates $w$ then
                    $g_r.N_r(w).N_u \leftarrow g_r.N_r(w).N_u \bigcup g_r.N_r(v).N_u$;
if $seeds_c \neq \emptyset$ then
    for each $s \in seeds_c$ do
        for each $I \in \mathcal{I}(s)$ do
            $N_u(I) = N_u(I) \bigcup g_r.N_r(s).N_u$;
    $g_r.t_c \leftarrow \sum_{I_j \in \mathcal{I}, j \neq i} |N_u(I_j)|$;
Return $g_r$;
```

Theorem 5.1. *Given the competitive seed sets $S_c = \{S_j | j \neq i, 1 \leq j \leq |\mathcal{I}|\}$, the adoption $f(S_i)$ of the influence I_i on a seed set S_i under the C-IC model is n time the expected adoption probability $E[h(u, I_i)]$, where u is an arbitrary node in the influence graph G.*

A variable $P(u, I_i)$ is defined as the adoption probability with which an arbitrary node $u \in G$ adopts I_i. Hence, we have $P(u, I_i) = E[h(u, I_i)]$ and $f(S_i) = n \cdot P(u, I_i)$ according to Theorem 5.1. The S-AM problem under the C-IC model is equivalent to identifying k nodes to maximize $P(u, I_i)$.

5.2. Reverse multiple influence sampling

$P(u, I_i)$ is estimated based on the following ideas. A random variable $p(u, I_i)$ is defined as the possibility with which a uniformly randomly selected node $u \in G$ adopts I_i in a random diffusion process. Thus, if we have sufficient instances of $p(u, I_i)$, we can approximate $P(u, I_i)$ with the mean of these instances due to $P(u, I_i) = E[p(u, I_i)]$. To achieve this goal, the RMIS algorithm is proposed to generate an instance of $p(u, I_i)$. We first select a node u uniformly at random from the reverse influence graph G^T which is the transpose graph of G. Then, an RMIS is conducted in G^T as presented in Algorithm 1 and outputs a structure g_r. Finally, we exploit g_r to calculate an instance of $p(u, I_i)$, denoted by $p_{g_r}(u, I_i)$, as described in Section 5.2.1.

To facilitate understanding of Algorithm 1, some important notations and their meanings are listed in Table 2. As described in Algorithm 1, an RMIS proceeds like a random breadth-first traversal from u. First of all, we identify whether u is a competitive seed or not (Lines 7-9). If not, we put u into the set $next$ to start searching for nodes which are able to influence it (Line 10). In the loop (Lines 14-29), each out-neighbor of each node in $active$ is attempted to activate (Lines 17 and 29) with the influence probability between the two nodes. The RMIS ends after a loop ends if no nodes are activated or competitive seeds are activated in the loop (Line 11). Activating the competitive seeds causes the RMIS to terminate early because the nodes activated in subsequent loops cannot influence u. Because the RMIS is the reverse process of influence diffusion under the C-IC model. Unlike the breadth-first traversal, a node can be activated more than once by different nodes in the same loop (Lines 27-29). This case corresponds to a node that can activate multiple out-neighbors in G under the C-IC model. Furthermore, that a node is activated by different nodes in the RMIS process means it can spread influences to u along different paths. Correspondingly, N_u of such a node is set to the union of N_u of the nodes that activate it (Line 29). To facilitate the computation of $p(u, I_i)$, we compute the total number of times u receives competitive influences, denoted by $g_r.t_c$, after the RMIS process ends (Lines 30-34).

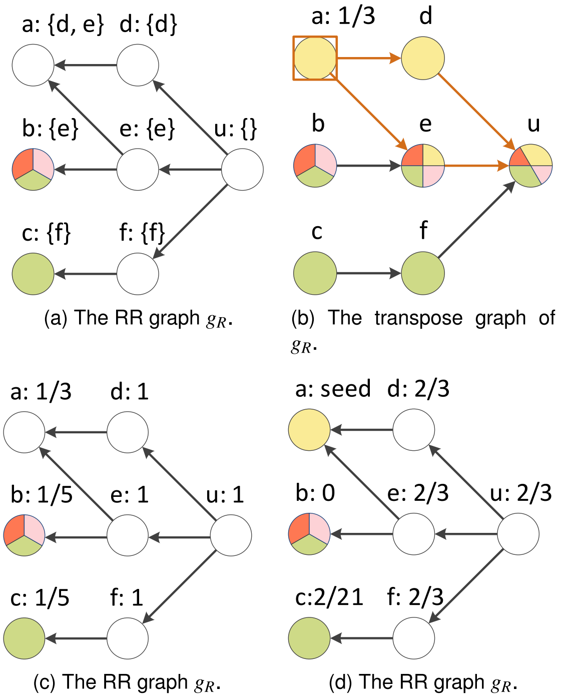
Example. Fig. 3(a) exhibits an RMIS process from a randomly selected node u. Since u is not a competitive seed, the RMIS process searches the nodes that can influence it until the competitive seeds (b and c) are activated. If an activated node v activates another node w, we add a new structure n_r corresponding to w into g_r. Note that the node set next to the symbol of each node in Fig. 3(a) is its N_u. For example, a's $N_u = \{d, e\}$ is the union of d's $N_u = \{d\}$ and e's $N_u = \{e\}$, since a is activated by d and e. It means that a can influence g via d and e.

5.2.1. Adoption probability computation

Logically, the activated nodes and activated edges in an RMIS process make up a reverse reachable (RR) graph, denoted by g_R, which is an un-weighted directed graph. In g_R, if there is a path from u to a node w, w is able to spread influences to u. Let g be the transpose graph of g_R and we calculate the adoption probability, denoted by $p_{g_R}(u, I_i)$, of u on g_R by $H(N_A(u, I_i))$ that is the probability of u adopting I_i in g.

In order to calculate $p_{g_R}(u, I_i)$, we need to calculate the number of times u receives each influence, i.e., the number of neighbors sending each influence to u. Any node sends all the influences it has received to the nodes it activates under the C-IC model. Besides, if there are paths from an activated neighbor n_u of u to a node w in g_r, n_u is added into w's N_u in Algorithm 1. Hence, each seed can send all the influences to u through the nodes in its N_u. As a result, given the seeds, we can get the neighbors sending each influence to u and compute the number of times u receives competitive influences (Lines 31-34 in Algorithm 1).

Example. Fig. 3(b) exhibits how the influences spread in the transpose graph of g_R where a is I_i's seed (yellow), and b and c are the competitive seeds. I_i is spread to u twice via d and e, and the competitive influences are sent to u four times via e and f. Therefore, u will adopt influence I_i with the probability $p_{g_R}(u, I_i) = 1/3$.



(a) The RR graph g_R.
(b) The transpose graph of g_R.
(c) The RR graph g_R.
(d) The RR graph g_R.

Fig. 3. Illustrate RMIS and the adoption probability computation.

5.3. Seed selection

Algorithm 1 outputs g_r that corresponds to an RR graph g_R. Denoted by $\Gamma_{g_r}(S_i)$ the adoption of S_i on u in g_r and let $\Gamma_{g_r}(S_i) = p_{g_R}(u, I_i)$, where u is the sampling source of g_r and S_i is I_i's seed set. Denoted by S_{i,g_r} the set of I_i's seeds in g_r and by $N_u(I_i)$ the set of u's out-neighbors spreading I_i to u. Hence we have $N_u(I_i) = \bigcup_{v \in S_{i,g_r}} g_r.N_r(v).N_u$, where $g_r.N_r(v).N_u$ is N_u of $g_r.N_r(v)$. We calculate the adoption gain $\Gamma_{g_r}(S_i)$ as

$$\Gamma_{g_r}(S_i) = \begin{cases} 1, & \text{if } p_{S_{i,g_r}} < g_r.d_c; \\ \frac{|N_u(I_i)|}{|N_u(I_i)|+g_r.t_c}, & \text{if } p_{S_{i,g_r}} = g_r.d_c, \end{cases}$$

where $g_r.t_c$ is the number of times u receive the competitive influences and $p_{S_{i,g_r}}$ is the shortest distance from u to S_{i,g_r}.

Denote by $\mathcal{G}_r$ the set consisting of large numbers of outputs of Algorithm 1. The adoption of I_i on a seed set S_i on the outputs in $\mathcal{G}_r$, denoted by $\Gamma_{\mathcal{G}_r}(S_i)$, is calculated by $\Gamma_{\mathcal{G}_r}(S_i) = \Sigma_{g_r \in \mathcal{G}_r} \Gamma_{g_r}(S_i)/|\mathcal{G}_r|$. We have the following conclusion.

Theorem 5.2. *Given the set of competitive seed sets $\mathcal{S}_c = \{S_j | j \neq i, 1 \leq j \leq |\mathcal{I}|\}$, the adoption $\Gamma_{\mathcal{G}_r}(S_i)$ of the influence I_i based on $\mathcal{G}_r$ is non-negative, monotone, and submodular.*

In order to find a seed set of size k with a maximum adoption based on $\mathcal{G}_r$, we develop a Seed-Selection algorithm in Algorithm 2. In Algorithm 2, the marginal gain of each node is initialized before selecting the first seed (Lines 3-5). For a node v in an output g_r, assume it is a new seed and calculate the marginal gain on g_r by $\Gamma_{g_r}(\{v\} \bigcup S_i) - \Gamma_{g_r}(S_i)$. Then we calculate the marginal gain of the node by accumulating all marginal gains of it on the outputs in $\mathcal{G}_r$. The seeds are selected one by one until the number of seeds reaches k (Lines 6-12). In each loop, the node providing the maximum marginal gain is chosen as a new seed s (Line 7). Then we update the marginal gains of the nodes in the outputs that contain s (Lines 9-11). For a node in an output g_r, the new marginal gain on g_r should be calculated based on $\{s\} \bigcup S_i$. Thus, we update the marginal gain by subtracting the original marginal gain based on S_i and then adding the new marginal gain based on $\{s\} \bigcup S_i$. Finally, we output S_i and $\Gamma_{\mathcal{G}_r}(S_i)$ which is the adoption of I_i (Line 14).

Algorithm 2 Seed-Selection

```
Input: A set of outputs of Algorithm 1 $\mathcal{G}_r$, an integer $k$
Output: A seed set $S_i$ of the influence $I_i$ and the adoption $\Gamma_{\mathcal{G}_r}(S_i)$
 1: $S_i \leftarrow \emptyset$ and $\Gamma_{\mathcal{G}_r}(S_i) \leftarrow 0$;
 2: Set the marginal gain $\gamma(v)$ to be 0 for each node $v$ in $V$;
 3: for each $g_r$ in $\mathcal{G}_r$ do
 4:     for each $v$ in $g_r$ do
 5:         $\gamma(v) \leftarrow \gamma(v) + \Gamma_{g_r}(\{v\} \bigcup S_i) - \Gamma_{g_r}(S_i)$;
 6: for $i = 1$ to $k$ do
 7:     $s \leftarrow \arg\max_{\{u \in V - S_i\}} \gamma(u)$;
 8:     for each $g_r$ containing $s$ do
 9:         for each $v$ in $g_r$ and $v \neq s$ do
10:             $\gamma(v) \leftarrow \gamma(v) - (\Gamma_{g_r}(\{v\} \bigcup S_i) - \Gamma_{g_r}(S_i))$;
11:             $\gamma(v) \leftarrow \gamma(v) + (\Gamma_{g_r}(\{v\} \bigcup \{s\} \bigcup S_i) - \Gamma_{g_r}(\{s\} \bigcup S_i))$;
12:     $\Gamma_{\mathcal{G}_r}(S_i) \leftarrow \Gamma_{\mathcal{G}_r}(S_i) + \gamma(s)$;
13:     $S_i \leftarrow S_i \bigcup \{s\}$;
14: Return $S_i$ and $\Gamma_{\mathcal{G}_r}(S_i)$;
```

Theorem 5.3. *Let S_i^+ be the seed set output by Algorithm 2. Let S_i^* be an optimal seed set with maximum adoption $\Gamma_{\mathcal{G}_r}(S_i)$ over all sets of size k on the output set $\mathcal{G}_r$. $\Gamma_{\mathcal{G}_r}(S_i^+) \geqslant (1 - 1/e) \cdot \Gamma_{\mathcal{G}_r}(S_i^*)$ holds.*

Example. Fig. 3(c) and Fig. 3(d) explain how to initialize and update the marginal gain of each node in g_R, respectively. Let $\gamma_{g_R}(v)$ denote the marginal gain of a node v in g_R. In Fig. 3(c), there are two competitive seeds, b and c, and there are no seeds of the influence I_i. We compute the marginal gain of each node based on the assumption that it is a new seed. For instance, we have $\gamma_{g_R}(u) = 1$ since if u is the seed of I_i, u must adopt I_i. If e is the seed of I_i, we get $\gamma_{g_R}(e) = 1$ because I_i will reach u earlier than the competitive influences from b and c. Besides, assuming c is a seed of I_i, c spreads two influences to u and b spreads three influences to u, respectively. In the case, $\gamma_{g_R}(c) = 1/5$ is because u only receives I_i once from f

In Fig. 3(d), we assume a has been chosen to be a seed of I_i and the marginal gains of nodes in g_R need updating. Fig. 3(d) presents the updated marginal gain of each node and these marginal gains decrease compared to that in Fig. 3(c). In this case, we adjust the value of the marginal gain of a node v by using $\gamma_{g_R}(v) = \Gamma_{g_R}(\{v\} \bigcup \{a\}) - \Gamma_{g_R}(\{a\})$. For example, $\gamma_{g_R}(d) = 2/3$, since $\gamma_{g_R}(d) = \Gamma_{g_R}(\{d\} \bigcup \{a\}) - \Gamma_{g_R}(\{a\}) = 1 - 1/3 = 2/3$. And we find $\gamma_{g_R}(b) = 0$ because even if b is used as a new seed of I_i, the number of times u receives I_i keeps the same. In addition, if c is used as a new seed of I_i, we have $\gamma_{g_R}(c) = \Gamma_{g_R}(\{c\} \bigcup \{a\}) - \Gamma_{g_R}(\{a\}) = 3/7 - 1/3 = 2/21$

5.4. RAE+DSSA algorithm

The DSSA [11] algorithm, one of the state-of-the-art algorithms, provides a $(1 - 1/e - \epsilon)$-approximate solution with a probability of at least $(1 - \delta)$ under the IC model. It strives to improve the efficiency of tackling the IM problem while ensuring the approximation guarantee and reliability of the solution.

The RAE method and the Seed-Selection algorithm bridge the gap between the C-IC model and the DSSA algorithm. We adapt DSSA to fit the C-IC model that models multiple competitive influences spreading simultaneously in a social network. We exploit the RAE method to compute adoptions based on $\mathcal{G}_r$ in the changed DSSA algorithm. We replace the RIS technology with the RMIS technology to generate samples for estimating adoption and replace the Max-Coverage algorithm in the DSSA algorithm with the Seed-Selection algorithm to provide the seed set with the maximum adoption and its adoption. The changed algorithm is called RAE+DSSA.

Since the Seed-Selection algorithm provides a $(1 - 1/e)$-approximate solution based on $\mathcal{G}_r$. According to [11], we have the following conclusion: Given $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$, RAE+DSSA returns an $(1 - 1/e - \epsilon)$-approximate optimal solution with probability at least $(1 - \delta)$.

6. One-hop adoption estimation

To reduce the time and space complexity of estimating adoption, only seeds and their out-neighbors are taken into consideration. The adoption of an influence is defined as the sum of the adoption gains (the adoption probabilities of the influence) on its seeds and their out-neighbors.

Denote by $g(S_i, v_0)$ the adoption gain that a specific influence I_i can obtain on a node v_0 by using the seed set S_i. v_0 may be in one of the following three states. (1) If v_0 is a seed of influence I_i, we have $g(S_i, v_0) = \frac{1}{|\mathcal{I}(v_0)|}$. (2) If v_0 is not a seed of influence I_i but v_0 is a seed of any competitive influence, we have $g(S_i, v_0) = 0$. (3) If v_0 is not a seed of any influence, we can compute $g(S_i, v_0)$ by using Eq. (2). In this case, we consider an in-coming star graph $G_{in}(v_0) = (V', E', p')$ which is a subgraph of G. V' includes v_0 and its in-neighbors, i.e., $V' = \{v_0\} \bigcup N_{in}(v_0)$. E' includes all the edges from the in-neighbors to v_0.

Denote by T and S the seed set of all the influences and I_i in $N_{in}(v_0)$, respectively, i.e., $S = S_i \bigcap N_{in}(v_0)$ and $T = (\bigcup_{j=1}^{|\mathcal{I}|} S_j) \bigcap N_{in}(v_0)$. Denote by $\mathcal{T}$ the set consisting of all subsets of T. We have

$$g(S_i, v_0) = \sum_{T' \in \mathcal{T}} \prod_{v \in T'} p_v \cdot \prod_{v \in T - T'} (1 - p_v) \cdot \frac{|S \cap T'|}{\sum_{v \in T'} |\mathcal{I}(v)|}, \tag{2}$$

where p_v is the probability with which v activates v_0.

The time complexity of computing $g(S_i, v_0)$ by using Eq. (2) is $\Omega(2^{|T|})$. However, Eq. (2) only computes an adoption gain on just one node. Therefore, an approximation of the adoption gain is imperative to reduce the time complexity. To simplify the computation, we first transform Eq. (2). Let set $C = T - S$ be the seed set in which seeds only serve for the competitive influences. Denote by $\mathcal{S}$ and $\mathcal{C}$ the set consisting of all subsets of S and C, respectively. Since any subset of T can be uniquely divided into two subsets of S and C. We have

$$g(S_i, v_0) = \sum_{S' \in \mathcal{S}} \sum_{C' \in \mathcal{C}} \left\{ \prod_{v \in S'} p_v \cdot \prod_{v \in C'} p_v \cdot \prod_{v \in S - S'} (1 - p_v) \right. \\ \left. \cdot \prod_{v \in C - C'} (1 - p_v) \cdot \frac{|S'|}{\sum_{v \in S'} |\mathcal{I}(v)| + \sum_{v \in C'} |\mathcal{I}(v)|} \right\}.$$

Since the in-neighbors activate v_0 independently in the C-IC model, the adoption gain can be computed as

$$g(S_i, v_0) = \sum_{S' \in \mathcal{S}} \left\{ \prod_{v \in S'} p_v \cdot \prod_{v \in S - S'} (1 - p_v) \cdot \sum_{C' \in \mathcal{C}} \prod_{v \in C'} p_v \right. \\ \left. \cdot \prod_{v \in C - C'} (1 - p_v) \cdot \frac{|S'|}{\sum_{v \in S'} |\mathcal{I}(v)| + \sum_{v \in C'} |\mathcal{I}(v)|} \right\}.$$

Since the influence probabilities are small in C-IC models, we replace $1 - p_v$ in $\prod_{v \in C - C'} (1 - p_v)$ with 1 and neglect terms with $C' \neq \emptyset$ in the above equation and get

$$\sum_{C' \in \mathcal{C}} \prod_{v \in C'} p_v \cdot \prod_{v \in C - C'} (1 - p_v) \cdot \frac{|S'|}{\sum_{v \in S'} |\mathcal{I}(v)| + \sum_{v \in C'} |\mathcal{I}(v)|} \approx \frac{|S'|}{\sum_{v \in S'} |\mathcal{I}(v)|}.$$

Then, we further simplify the computation by only considering terms with $|S'| = 1$ and finally approximate $g(S_i, v_0)$ as

$$\begin{aligned} g(S_i, v_0) &\approx \sum_{S' \in \mathcal{S}} \prod_{v \in S'} p_v \cdot \prod_{v \in S - S'} (1 - p_v) \cdot \frac{|S'|}{\sum_{v \in S'} |\mathcal{I}(v)|} \\ &\approx \sum_{u \in S} p_u \cdot \prod_{v \in S - \{u\}} (1 - p_v) \cdot \frac{1}{|\mathcal{I}(u)|} \end{aligned} \tag{3}$$

Suppose $s \in N_{in}(v_0)$ is a candidate seed of I_i and S is the current seed set of I_i in $N_{in}(v_0)$. Based on Eq. (3), the marginal gain of s on v_0, denoted by $\Delta(s, S, v_0)$, is estimated as

$$\begin{aligned} \Delta(s, S_i, v_0) &= g(S_i \bigcup \{s\}, v_0) - g(S_i, v_0) \\ &= p_s \cdot \prod_{v \in S} (1 - p_v) \cdot \frac{1}{|\mathcal{I}(s)|} - p_s \cdot g(S_i, v_0). \end{aligned} \tag{4}$$

Table 3
Datasets.

Dataset	n	m	Type
NetHEPT	15.2K	31.4K	Undirected
NetPHY	37.1K	174.2K	Undirected
Enron	36.7K	367.7K	Directed
Epinions	131.6K	840.8K	Directed
DBLP	0.6M	2.0M	Undirected
Pokec	1.6M	30.6M	Directed
Orkut	3.0M	117.2M	Undirected
LiveJournal	4.8M	68.5M	Directed

Based on Eq. (4), the total marginal gain $\mathcal{F}(s, S_i)$ of s is estimated as the sum of marginal gains on s and its out-neighbors. Intuitively, the adoption gain on s is defined as $\Delta(s, S_i, s) = \frac{1}{|\mathcal{I}(s)|} - g(S_i, s)$. Therefore, we have

$$\mathcal{F}(s, S_i) = \Delta(s, S_i, s) + \sum_{v \in N_{out}(s), v \notin S_C} \Delta(s, S_i, v). \tag{5}$$

A greedy strategy is used to select the seeds with maximum marginal gain iteratively until the number of seeds reaches k in Algorithm 3. In each loop, the marginal gains of two classes of nodes change after adding a new seed s into S_i and need to be updated. One class includes nodes that share common out-neighbors with s. The marginal gains of those nodes obtained from the common out-neighbors will reduce after adding s into S_i according to Eq. (4) and Eq. (5). The other class includes the in-neighbors of s since they cannot activate s and obtain marginal gains from s. Note that we only update the marginal gains that will change after selecting a new seed instead of re-computing the marginal gains of all nodes (Line 6). Therefore, the running time of Algorithm 3 is further reduced.

Algorithm 3 GOAE

```
Input: Seed number $k$, influence graph $G = (V, E, p)$, seed sets of the
       competitive influences
Output: A seed set $S_i$ of the influence $I_i$
1: Let seed set $S_i = \emptyset$;
2: Initialize the marginal gain $\mathcal{F}(u, S_i)$ for every node $u \in V$;
3: for $i = 1$ to $k$ do
4:     $s \leftarrow \arg\max_{\{u \in V - S_i\}} \mathcal{F}(u, S_i)$;
5:     Add $s$ into $S_i$;
6:     Update $\mathcal{F}(u, S_i)$ of nodes whose marginal gains change after adding $s$
       into $S_i$;
7: Return $S_i$;
```

7. Experiments

Numerous experiments are conducted to evaluate the performance of RAE+DSSA and GOAE in this section. All the experiments are conducted on a Linux machine with 2.4 GHz Intel Xeon E5-2680 v4 and 251.6 GB memory. The algorithms tested are implemented in C++ and compiled with g++ 4.8.5.

7.1. Experimental setup

Datasets: Seven real network datasets are used in the experiments, as listed in Table 3. NetHEPT, NetPHY, and DBLP are three collaboration networks, which are downloaded from [46]. Enron is an email communication network and Epinions, Pokec, Orkut, and LiveJournal are online social networks, which are downloaded from SNAP [47].

Influence probability settings. We adopt the following three classic models to set the influence probabilities.

UC [12]: The influence probabilities of all the edges are uniformly set to 0.1.

WC [12]: The multiplicity of edges is considered in the WC model. For an edge (u, v), we set the influence probability $p_{uv} = c_{uv}/d_{in}(v)$, where c_{uv} is the number of the parallel edges from u to v and $d_{in}(v)$ is the in-degree of v. The WC model is asymmetric.

TC [15]: The influence probability of each edge is selected uniformly at random from the following three values 0.001, 0.01, and 0.1. The three values reflect the different strength of influence from weak to strong.

Size of seed set. Although there is no limit to the number of competitive influences in the C-IC model, we consider two competitive influences in the experiments. Two settings of the seed number k are considered: (i) $k \in \{1, 10, 20, 30, 40, 50\}$, referred to as the small k setting. Meanwhile, the sizes of the two competitive seed sets are set to twenty. We first assign the top ten largest out-degree nodes to the two competitive influences as common seeds. Then we select seeds from the remaining nodes and assign them to the two competitive influences as follows. Step (a), the node with the largest out-degree is assigned to one competitive influence and the other competitive influence gets the node with the second largest out-degree. Step (b), exchange the order in which we assign the selected seeds. Step (c), return to Step (a) until the competitive influences both have twenty seeds. (ii) $k \in \{1, 5000, 10000, 15000, 20000, 25000\}$, referred to as the large k setting. Moreover, the sizes of the two competitive seed sets are set to ten thousand. They have two thousand five hundred common seeds whose out-degree are in the top two thousand five hundred. The other seeds of the two competitive influences are assigned as in the small k setting.

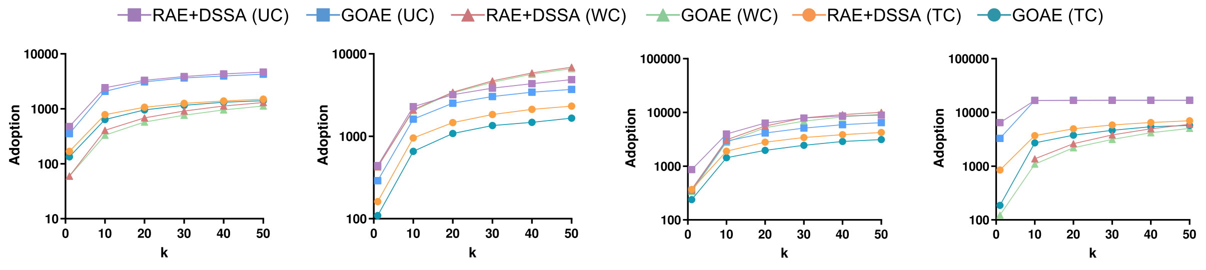
Parameter settings. We implement a sequential version of the D-SSA-fix algorithm [11] and integrate the RMIS algorithm and the Seed-Selection algorithm into it for the S-AM problem under the C-IC model. It removes the influence of parallel sampling in the original version on computational performance. When not confusing, DSSA refers to our implementation in this section. In all the experiments, we set $\epsilon = 0.1$ and $\delta = 0.1$, and retain the settings of other parameters of the D-SSA-fix algorithm in [11]. For each experiment, we run each algorithm five times and report the mean of the measurements. To evaluate and compare the quality of the solutions returned by the two algorithms, we run one thousand diffusion simulations for each seed set and report the average adoption as its adoption.

7.2. Experimental results under small k setting

We experimentally evaluate RAE+DSSA and GOAE under the small k setting on NetHEPT, NetPHY, Enron, Epinions, and DBLP. The small k setting is a classic setting of the size of seed sets, which is used in most existing work on the IM problem. We compare the solution quality, the running time, and the memory usage of the two algorithms under the C-IC model with the three influence probability settings. The results of RAE+DSSA under the WC setting when $k = 1$ are not reported since the running time is over 5 h.

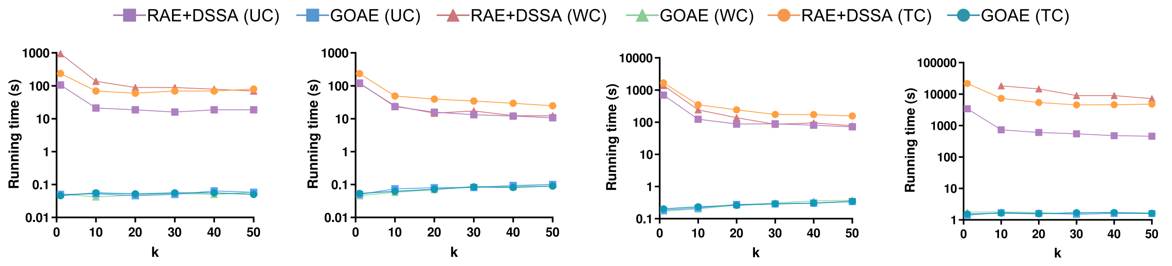
Solution quality. Fig. 4 shows the adoptions of seed sets returned by RAE+DSSA and GOAE on the four real-world networks. The adoptions increase with the size of the seed set in all experiments on the four networks. The adoptions of most seed sets returned by GOAE are not significantly less than those of seed sets returned by RAE+DSSA. It illustrates that GOAE returns seed sets of comparable quality to RAE+DSSA. We additionally report the proportions of the adoptions of seed sets returned by GOAE to the adoptions of seed sets returned by RAE+DSSA in Table 4. The results also show that the quality of most solutions returned by the two algorithms is comparable. Most (85%) of the proportions are larger than 70%. In some cases, the proportion even reaches 1. It verifies that the adoption estimated by using Eq. (3) is a good indicator of seed set quality.

Computation cost. Fig. 5 and Fig. 6 show the running time and memory usage of RAE+DSSA and GOAE on the four real-world networks, respectively. The computation cost, both running time and memory usage, of RAE+DSSA are mainly affected by the number of samples and the computation cost of each sample. The computation



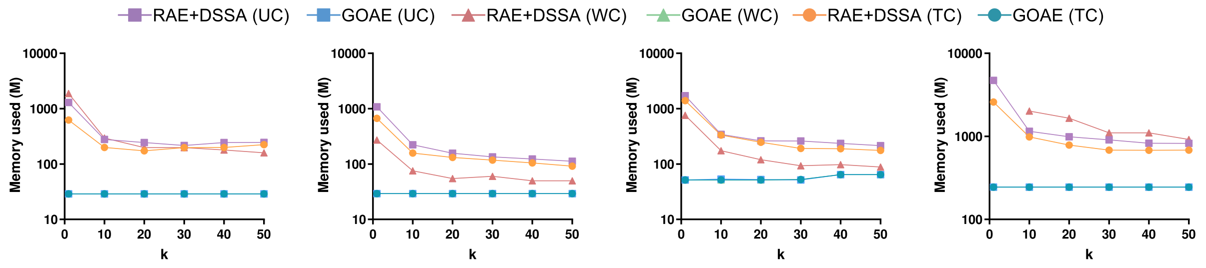
(a) NetPHY (b) Enron (c) Epinions (d) DBLP

Fig. 4. Adoption vs. k under the three influence probability settings on four real-world networks.



(a) NetPHY (b) Enron (c) Epinions (d) DBLP

Fig. 5. Running time vs. k under the three influence probability settings on four real-world networks.



(a) NetPHY (b) Enron (c) Epinions (d) DBLP

Fig. 6. Memory used vs. k under the three influence probability settings on four real-world networks.

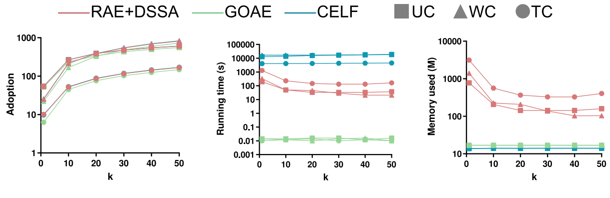
Table 4
The proportions of the adoptions of seed sets obtained by GOAE to the adoptions of seed sets obtained by RAE+DSSA.

	WC						UC						TC					
k	1	10	20	30	40	50	1	10	20	30	40	50	1	10	20	30	40	50
NetHEPT	0.88	0.78	0.86	0.89	0.87	0.90	0.95	0.90	0.87	0.91	0.91	0.92	0.63	0.87	0.88	0.88	0.88	0.89
NetPHY	1.00	0.82	0.84	0.84	0.85	0.86	0.74	0.86	0.93	0.94	0.91	0.92	0.80	0.81	0.89	0.92	0.95	0.94
Enron	1.00	0.98	0.98	0.96	0.97	0.96	0.66	0.71	0.79	0.79	0.79	0.76	0.68	0.69	0.74	0.73	0.70	0.72
Epinions	0.91	0.90	0.93	0.87	0.90	0.92	0.41	0.74	0.65	0.65	0.69	0.72	0.65	0.76	0.70	0.72	0.74	0.74
DBLP	–	0.80	0.84	0.82	0.85	0.84	0.51	1.00	1.00	1.00	1.00	1.00	0.22	0.74	0.76	0.79	0.83	0.82

cost of each sample is determined by the RAE method and the number of samples is determined by DSSA. The computation costs of RAE+DSSA when $k = 1$ are significantly higher than the computation costs under other values of k since DSSA needs large numbers of samples to provide an approximation guarantee for the solution when the adoption of a candidate solution is small. The adoptions of the candidate solutions increase with k, which results in a reduction in the number of samples required for DSSA. Therefore, the computation cost of RAE+DSSA decreases when k increases. Note that when k reaches a certain value, this trend will end, after which the computation cost of RAE+DSSA will increase with k to provide an approximation guarantee for the solution. On the contrary, the computation cost of GOAE is not only significantly lower than the computation cost of RAE+DSSA but also grows very slowly with k. Fig. 5 shows GOAE runs at least two orders of magnitude faster than RAE+DSSA. Furthermore, GOAE runs up to four orders of magnitude faster than RAE+DSSA under the three influence probability settings. RAE+DSSA is very time-consuming for large networks, especially when $k = 1$. In Fig. 6, the memory usage of RAE+DSSA is at most two orders of magnitude larger than that of GOAE. Besides, the computation cost of RAE+DSSA is significantly influenced by the influence probability settings, while that of GOAE are very closed under the three influence probability settings. For the same k, the computation cost of RAE+DSSA varies, because the adoption of the solution is affected by the three influence probability settings. Like the previous analysis, if the adoption of the solution is less, e.g., under the TC setting, RAE+DSSA will need more samples.

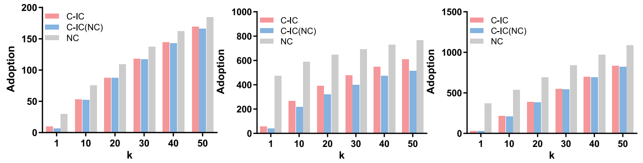
Table 5
Adoptions of seed sets obtained by GOAE under the three influence probability settings on Pokec.

	Adoption (k)															
k	1	100	200	300	400	500	600	700	800	900	1000	2000	3000	4000	5000	10 000
TC	0.47	3.67	5.83	7.87	9.18	10.14	11.13	12.15	13.18	14.17	15.21	18.74	19.74	20.74	21.74	26.74
UC	1.92	8.78	10.41	12.22	14.15	16.07	17.33	17.43	17.53	17.63	17.73	18.73	19.72	20.72	21.72	26.72
WC	0.51	8.84	12.86	16.66	17.14	17.23	17.31	17.40	17.49	17.59	17.68	18.63	19.61	20.60	21.59	26.56



(a) Adoption (b) Running time (c) Memory used

Fig. 7. Experiments on NetHEPT.



(a) TC (b) UC (c) WC

Fig. 8. Adoption on NetHEPT under the C-IC model and the non-conformity model.

7.2.1. Experimental results on NetHEPT

We also implement the CELF algorithm in [13], as the baseline method, and run it on NetHEPT for the S-AM problem. In the CELF algorithm, the adoptions are estimated based on influence diffusion simulations and we run ten thousand simulations for each candidate seed set. But we do not use it on other datasets, since it takes tens of hours for one experiment.

To compare with RAE+DSSA and GOAE, the experimental results are presented in Fig. 7. We report the adoption, running time, and used memory of the three methods. The three measurements of RAE+DSSA and GOAE show similar trends as shown in Fig. 4, 5, and 6. As shown in Fig. 7(a), RAE+DSSA achieves comparable adoptions with the CELF algorithm. The adoptions obtained by GOAE are slightly less than the adoptions obtained by RAE+DSSA and CELF. However, GOAE is three to five orders of magnitude faster than RAE+DSSA and five to six orders of magnitude faster than CELF on NetHEPT in Fig. 7(b). In addition, RAE+DSSA is up to three orders of magnitude faster than CELF on NetHEPT. Fig. 7(c) shows GOAE and CELF use nearly the same amount of memory which is much less than the memory used by RAE+DSSA. In summary, GOAE and RAE+DSSA outperform CELF, since running time is the main challenge of the S-AM problem.

To evaluate the C-IC model, we compare it with a non-conformity diffusion model and we also use the CELF algorithm under the non-conformity diffusion model on NetHEPT, under which a user selects one influence from her received influences with the same probabilities. For example, if a user receives three influences, and then she will adopt one of them with a one-third probability for each received influence.

The adoptions used three influence probabilities settings are shown in Fig. 8. In Fig. 8, C-IC and NC represent adoptions under the C-IC model and the non-conformity diffusion model respectively. C-IC(NC) stands for adoptions under the C-IC model, but the seed sets are selected under the non-conformity diffusion model. Fig. 8 shows that the adoptions under the non-conformity diffusion model are always larger than the adoptions under the C-IC model. Besides, the adoptions denoted by C-IC(NC) are consistently less than the adoptions under the C-IC model. It means that if a diffusion model ignores the role of conformity, the adoptions are overestimated and the selected seed sets are not the optimal. The difference in the adoptions between C-IC(NC) and C-IC in Figs. 8(a) and 8(c) is small and less than that in Fig. 8(b). Because the influence probabilities in TC are much smaller than UC and WC, which means that a user can only be activated by fewer neighbors and receive fewer influences. It weakens the role of conformity in the adoption stage and leads to a small difference in the adoptions between C-IC(NC) and C-IC. Similarly, a user can only be activated by a few neighbors under the WC settings, since the influence probabilities in WC are set as $p_{uv} = c_{uv}/d_{in}(v)$, which makes the expected number of neighbors activating a user is small. This results in the adoptions of C-IC(NC) and C-IC being close. While under the UC model, an activated user can receive more influences from more friends, which enables conformity to play a role in the adoption stage. Therefore, in Fig. 8(b), the adoptions under the C-IC model are significantly larger than the adoptions denoted by C-IC(NC). Due to the easy access to online content and the convenience of communication, a person in a social network often receives a wealth of messages both in variety and frequency from her friends. We believe the real information dissemination environments are closer to the UC influence probabilities setting. To summarize, compared with a non-conformity diffusion model, the C-IC model is more conducive to obtaining superior seed sets.

7.3. Experimental results under large k setting

We additionally evaluate GOAE algorithms under the large k setting on the three large real-world networks, i.e., Pokec, LiveJournal, and Orkut.

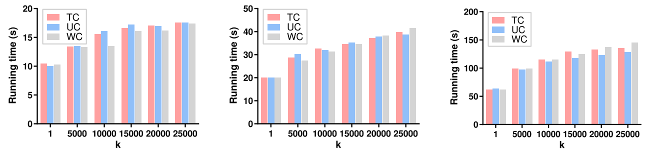
Adoption. We report the adoptions of solutions obtained by GOAE on Pokec in Table 5 varying k from 1 to 10000. Table 5 shows the adoption increases with k. Besides, the results indicate that only a small fraction of nodes can influence a large number of nodes. We observe that the adoption gain becomes almost equal to the number of new seeds added after this small fraction of nodes are selected as seeds. The phenomenon is consistent with the motivation of adoption maximization.

Computation cost. Fig. 9 and Table 6 show the running time and the memory usage of GOAE on the three large networks, respectively. Similar to the experimental results under the small k setting, the computation cost of GOAE is very low and increases very slowly with k. The results illustrate that GOAE scales well for large k values on large networks. For example, GOAE returns 25000 seeds within 150 s using less than 6 GB of memory on Orkut which has more than 117 million edges.

For each value of k, the running time is similar and the memory usage is almost the same under the three influence probability settings. It means that the computation cost of GOAE is not influenced by the influence probability setting. Furthermore, we observe that selecting one seed seems time-consuming compared to selecting a large number of seeds. This is because GOAE needs to calculate and sort the marginal gains of all the nodes before selecting the first seed.

8. Conclusion

In this paper, we propose the C-IC model to model influence diffusion and adoption in a competitive social network to obtain a more realistic and effective seed set. The adoption under the C-IC model is non-negative, monotone, and submodular. Furthermore, two AM



(a) Pokec (b) LiveJournal (c) Orkut

Fig. 9. Running time vs. k under the three influence probability settings on three real-world networks.

Table 6
Memory used vs. k under the three influence probability settings.

	Memory used (M)					
k	1	5000	10 000	15 000	20 000	25 000
Pokec	618	1050	1050	1050	1050	1050
LiveJournal	2600	2657	2657	2657	2657	2657
Orkut	2372	6039	6039	6039	6039	6039

problems, O-AM and S-AM, are proposed, which are both NP-hard. The RAE method based on RMIS is presented to estimate the adoption instead of the method based on influence diffusion simulation. Then it is integrated into the DSSA framework to obtain a solution to the S-AM problem with approximate guarantees. In addition, we propose the GOAE algorithm based on the OAE method for overcoming the challenge caused by the large-scale networks and large size of seed set. Experiments on eight real-world networks demonstrate the effectiveness of the two methods. Moreover, the GOAE algorithm runs up to four to five orders of magnitude faster than RAE+DSSA. The memory usage of GOAE is at most two orders of magnitude less than that of RAE+DSSA.

CRediT authorship contribution statement

Yonggang Liu: Formal analysis, Methodology, Software, Writing – original draft, Writing – review & editing. **Yikun Hu:** Project administration. **Siyang Yu:** Project administration. **Xu Zhou:** Supervision, Writing – review & editing. **Keqin Li:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We thank the reviewers for their constructive comments. The research was partially funded by the National Key R&D Program of China (Grant No. 2020YFB2104000), the Programs of NSFC (Grant Nos. 62172151, 62172157, 62172146), and the Programs of Natural Science Foundation of Hunan Province, China (Grant Nos. 2022JJ30009, 2023JJ10016).

Appendix

Proof of Corollary 4.1. Monotonicity and submodularity of $h(u, I_i)$ are proofed as follows.

Monotonicity of $h(u, I_i)$: For any node $u \in V$, if v is a new in-neighbor sending I_i to u, we have

$$\begin{aligned} &H(N_A(u, I_i) \cup \{v\}) - H(N_A(u, I_i)) \\ &= \frac{\sum_{I_j \in \mathcal{I}, I_j \neq I_i} |N_A(u, I_j)|}{(\sum_{I_j \in \mathcal{I}} |N_A(u, I_j)|)(\sum_{I_j \in \mathcal{I}} |N_A(u, I_j)| + 1)} \\ &= \frac{T}{(|N_A(u, I_i)| + T)[(|N_A(u, I_i)| + 1) + T]} \\ &\geq 0, \end{aligned} \tag{A.1}$$

where $T = \sum_{I_j \in \mathcal{I}, I_j \neq I_i} |N_A(u, I_j)| \geq 0$. Based on Eq. (A.1), $H(X) \leq H(Y)$ for any two sets $X \subseteq Y \subseteq N_{in}(u)$. Therefore, $H(N_A(u, I_i))$ is monotone increasing. This is consistent with Observation one.

Submodularity of $h(u, I_i)$: For any $X \subseteq Y \subseteq N_{in}(u)$ and $v \in N_{in}(u) - Y$, we have

$$\begin{aligned} &H(X \cup \{v\}) - H(X) \\ &= \frac{T}{(|X| + T)[(|X| + 1) + T]} \\ &\geq \frac{T}{(|Y| + T)[(|Y| + 1) + T]} \\ &= H(Y \cup \{v\}) - H(Y), \end{aligned}$$

since $|Y| \geq |X|$, where $T = \sum_{I_j \in \mathcal{I}, I_j \neq I_i} |N_A(u, I_j)|$. Therefore, $H(N_A(u, I_i))$ is submodular. This is consistent with Observation two. □

Proof of Corollary 4.2.. When $W = \emptyset$, $|W| = 0$, Eq. (1) is established.

For any set W, $|W| \geq 0$, if $H(X \cup W) - H(X) \geq H(Y \cup W) - H(Y)$ holds. Then, add a new element w into W. We have

$$\begin{aligned} &H(X \cup W \cup \{v\}) - H(X) \\ &= H(X \cup W \cup \{v\}) - H(X \cup W) + H(X \cup W) - H(X) \\ &\geq H(Y \cup W \cup \{v\}) - H(Y \cup W) + H(Y \cup W) - H(Y) \\ &= H(Y \cup W \cup \{v\}) - H(Y), \end{aligned}$$

because $H(X \cup W \cup \{v\}) - H(X \cup W) \geq H(Y \cup W \cup \{v\}) - H(Y \cup W)$ and $H(X \cup W) - H(X) \geq H(Y \cup W) - H(Y)$. This corollary is proofed. □

Proof of Theorem 4.3. Given the seed set $S = \bigcup_{i=1}^{|\mathcal{I}|} S_i$, the overall adoption $\mathcal{F}(S)$ under the C-IC model is equal to $\sigma(S)$ under the IC model. Thus, $\mathcal{F}(S)$ is non-negative, monotone, and submodular. Moreover, $\mathcal{F}(S)$ achieves the maximum value when the size of S reaches maximum, which needs seed sets of all the influences do not overlap. Theorem 4.3 is proved. □

Proof of Theorem 4.4. In G, each edge $e = (u, v)$ has a weight $p_e \in (0, 1]$ representing the probability that u activates v. The influence graph G is interpreted as a distribution $\mathcal{G}$ over determined unweighted instance graphs. An instance graph $g \in \mathcal{G}$ is a randomly generated graph where each edge e is independently removed from G with the probability $1 - p_e$. Our proof for the monotonicity and submodularity of $f(S_i)$ is based on the instance graphs in $\mathcal{G}$.

That a node in an instance graph is reachable from the seeds of an influence I_i means it is activated by I_i. A node may be activated by multiple influences and it will adopt only one influence from these influences. The adoption of I_i in an instance graph g is denoted by $f(g, S_i)$, where S_i is the seed set of I_i.

Monotonicity of $f(g, S_i)$: We first prove the monotonicity of $f(g, S_i)$ in an instance graph g. For a node v in g, denoted by $f_g(v, S_i)$ the adoption that I_i can obtain from v, i.e., $f_g(v, S_i) = h(v, I_i) = H(N_A(v, I_i))$. First of all, we analyze the adoption $f_g(v, S_i)$ in g, since $f(g, S_i) = \sum_{v \in V} f_g(v, S_i)$. Suppose a new seed $s \in V - S_i$ is added into S_i. $f_g(v, S_i)$ is monotone increasing, if $f_g(v, S_i \bigcup \{s\}) \geq f_g(v, S_i)$ holds for any S_i and s.

For the sake of simplicity, we first define three auxiliary variables, p_s, p_{S_i}, and p_{S_c}. They are the minimum distances from the new seed s, the seed set S_i, and the competitive seed set $S_c = \bigcup_{S_j \in S_c} S_j$ to v, respectively.

(1) If $v = s$, adding the new seed s into S_i may result in the following two cases.

(1.1) if $p_{S_c} = 0$, i.e. $v \in S_c$, we have $f_g(v, S_i \bigcup \{s\}) > 0$ and $f_g(v, S_i) = 0$ due to $p_{S_i} > p_{S_c}$.

(1.2) if $p_{S_c} > 0$, we have $f_g(v, S_i \bigcup \{s\}) = 1$ for $p_s = 0 < p_{S_c}$ and $f_g(v, S_i) \geq 0$.

Thus, given $v = s$, $f_g(v, S_i \bigcup \{s\}) - f_g(v, S_i) \geq 0$ holds.

(2) If $v \neq s$, there are more possible cases after adding s into S_i.

(2.1) if $p_{S_i} < p_{S_c}$, $f_g(v, S_i \bigcup \{s\}) = f_g(v, S_i) = 1$ holds no matter how long p_s is.

(2.2) if $p_{S_i} > p_{S_c}$, we have $f_g(v, S_i \bigcup \{s\}) \geq 0$ and $f_g(v, S_i) = 0$.

(2.3) if $p_{S_i} = p_{S_c}$, we have $0 < f_g(v, S_i) < 1$. (2.3.1) if $p_s < p_{S_i} = p_{S_c}$, $f_g(v, S_i \bigcup \{s\}) = 1$ holds; (2.3.2) if $p_s > p_{S_i} = p_{S_c}$, $f_g(v, S_i \bigcup \{s\}) = f_g(v, S_i)$ holds; (2.3.3) if $p_s = p_{S_i} = p_{S_c}$, $f_g(v, S_i \bigcup \{s\}) \geq f_g(v, S_i)$ holds due to the following analysis. Denote by $N_g(v, S_i)$ the set of v's in-neighbors that spread I_i to v, when S_i is used as the seed set of I_i. We have $|N_g(v, S_i)| \leq |N_g(v, S_i \bigcup \{s\})|$ for $N_g(v, S_i) \subseteq N_g(v, S_i \bigcup \{s\})$. Since the adoption probability function $h(u, I_i)$ is monotone increasing, we get $f_g(v, S_i \bigcup \{s\}) \geq f_g(v, S_i)$.

Thus, given $v \neq s$, $f_g(v, S_i \bigcup \{s\}) - f_g(v, S_i) \geq 0$ holds.

To summarize, $f(g, S_i)$ is monotone increasing owing to $f(g, S_i) = \sum_{v \in V} f_g(v, S_i)$.

Submodularity of $f(g, S_i)$: We prove the submodularity of $f(g, S_i)$ based on $f_g(v, S_i)$. Firstly, we aim to prove that

$$f_g(v, S_i \cup \{s\}) - f_g(v, S_i) \geq f_g(v, T_i \cup \{s\}) - f_g(v, T_i) \tag{A.2}$$

holds for any S_i and T_i, if $S_i \subseteq T_i \subseteq V$ and $s \in V - T_i$.

(1) If $v = s$, we have $f_g(v, T_i \cup \{s\}) = f_g(v, S_i \cup \{s\})$ owing to $s \in V - T_i$. Additionally, $f_g(v, T_i) \geq f_g(v, S_i)$ holds due to $T_i \supseteq S_i$. Thus, Eq. (A.2) holds.

(2)If $v \neq s$, there are more possible cases. We have $p_{T_i} \leq p_{S_i}$ because of $S_i \subseteq T_i$.

(2.1)If $p_s < p_{S_c}$, we have $f_g(v, T_i \cup \{s\}) = f_g(v, S_i \cup \{s\}) = 1$ and $f_g(v, T_i) \geq f_g(v, S_i)$. Eq. (A.2) holds.

(2.2)If $p_s > p_{S_c}$, we have $f_g(v, T_i \cup \{s\}) = f_g(v, T_i)$ and $f_g(v, S_i \cup \{s\}) = f_g(v, S_i)$. Eq. (A.2) holds.

(2.3) If $p_s = p_{S_c}$, there are three cases. (2.3.1) If $p_s = p_{S_c} < p_{T_i}$, we have $f_g(v, T_i \cup \{s\}) = f_g(v, S_i \cup \{s\})$ and $f_g(v, T_i) = f_g(v, S_i) = 0$. Eq. (A.2) holds. (2.3.2) If $p_s = p_{S_c} > p_{T_i}$, we have $f_g(v, T_i \cup \{s\}) - f_g(v, T_i) = 0$. In addition, we have $f_g(v, S_i \cup \{s\}) - f_g(v, S_i) \geq 0$. Thus, Eq. (A.2) holds. (2.3.3) If $p_s = p_{S_c} = p_{T_i}$, we consider v's in-neighbors spreading I_i to v. Denote by $N_g(v, S_i)$ the set of v's in-neighbors which spread I_i to v in g, when S_i is used as the seed set of I_i. Given $W = N_g(v, T_i \cup \{s\}) - N_g(v, T_i)$, we have $f_g(v, T_i \cup \{s\}) - f_g(v, T_i) = H(N_g(v, T_i) \cup W) - H(N_g(v, T_i)) \leq H(N_g(v, S_i) \cup W) - H(N_g(v, S_i))$ according to Corollary 4.2. In addition, we have $N_g(v, S_i) \subseteq N_g(v, T_i)$ because of $S_i \subseteq T_i$. Further, we have $W \subseteq N_g(v, S_i \cup \{s\}) - N_g(v, S_i)$ and $H(N_g(v, S_i \cup \{s\})) \geq H(N_g(v, S_i) \cup W)$ according to Corollary 4.1. Therefore, we have of $f_g(v, S_i \cup \{s\}) - f_g(v, S_i) = H(N_g(v, S_i \cup \{s\})) - H(N_g(v, S_i)) \geq H(N_g(v, S_i) \cup W) - H(N_g(v, S_i))$. As a result, Eq. (A.2) holds.

To summarize, $f(g, S_i)$ is submodular owing to $f(g, S_i) = \sum_{v \in V} f_g(v, S_i)$, i.e. $f(g, S_i \cup \{s\}) - f(g, S_i) \geq f(g, T_i \cup \{s\}) - f(g, T_i)$.

Monotonicity and submodularity of $f(S_i)$: Since $f(g, I_i)$ is monotone and submodular and $g \in \mathcal{G}$ is an instance graph of the influence graph G. The adoption $f(S_i)$ is monotone and submodular due to $f(S_i) = \sum_{g \in \mathcal{G}} Pr(g) \cdot f(g, S_i)$, where $Pr(g)$ is the probability of generating g. □

Proof of Theorem 4.5. Since computing $\sigma(S)$ under the IC model is #P-hard. In addition, computing $\mathcal{F}(S)$ under the C-IC model is equivalent to computing $\sigma(S)$ under the IC model. Therefore, the time complexity of computing $\mathcal{F}(S)$ is #P-hard. □

Proof of Theorem 4.6. The IC model is a special case of the C-IC model when $|\mathcal{I}| = 1$. Therefore, computing $f(S_i)$ under the C-IC model is not easier than computing $\sigma(S)$ under the IC model. Theorem 4.6 is proved. □

Proof of Theorem 5.1. Given the competitive seed sets, we have

$$\begin{aligned} f(S_i) &= \sum_{g \in \mathcal{G}} Pr(g) \cdot f(g, S_i) \\ &= \sum_{g \in \mathcal{G}} Pr(g) \cdot \sum_{u \in g} h(u, I_i) \\ &= \sum_{g \in \mathcal{G}} Pr(g) \cdot n \cdot E_{u \in g}[h(u, I_i)] \\ &= n \cdot E[h(u, I_i)]. \quad \square \end{aligned}$$

Proof of Theorem 5.2. Like $f_g(v, S_i)$ in the proof of Theorem 4.4, $\Gamma_{g_r}(S_i)$ is non-negative, monotone, and submodular. Theorem 5.2 is proved. □

Proof of Theorem 5.3. $\Gamma_{\mathcal{G}_r}(S_i)$ is non-negative, monotone, and submodular. Algorithm 2 is a greedy algorithm based on $\Gamma_{\mathcal{G}_r}(S_i)$. Therefore, the seed set S_i^+ output by Algorithm 2 provides a $(1 - 1/e)$-approximation of the optimal solution S_i^* on the output set $\mathcal{G}_r$. □

References

[1] D. Easley, J. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.

[2] E. Aronson, T.D. Wilson, R.M. Akert, Social Psychology, Pearson Education International, 2007.

[3] H. Li, P. Wu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey on parameter identification, state estimation and data analytics for lateral flow immunoassay: From systems science perspective, Internat. J. Systems Sci. 53 (16) (2022) 3556–3576.

[4] P. Wu, Z. Wang, B. Zheng, H. Li, F.E. Alsaadi, N. Zeng, AGGN: Attention-based glioma grading network with multi-scale feature extraction and multi-modal information fusion, Comput. Biol. Med. 152 (2023) 106457.

[5] J. Fang, Z. Wang, W. Liu, S. Lauria, N. Zeng, C. Prieto, F. Sikstrom, X. Liu, A new particle swarm optimization algorithm for outlier detection: Industrial data clustering in wire arc additive manufacturing, IEEE Trans. Autom. Sci. Eng. (2023) 1–14.

[6] S. Milgram, L. Bickman, L. Berkowitz, Note on the drawing power of crowds of different size, J. Personal. Soc. Psychol. 13 (2) (1969) 79–82.

[7] J.C. Coultas, K. Eriksson, Milgram revisited: Imitative behaviour is influenced by both the size and entitativity of the stimulus group, in: Annual British Psychological Society, 2014, pp. 137–146.

[8] A.C. Gallup, J.J. Hale, D.J.T. Sumpter, S. Garnier, A. Kacelnik, J.R. Krebs, I.D. Couzin, Visual attention and the acquisition of information in human crowds, Proc. Natl. Acad. Sci. 109 (19) (2012) 7245–7250.

[9] J. Egebark, M. Ekström, Liking what others "like": Using Facebook to identify determinants of conformity, Experimental Economics 21 (4) (2018) 793–814.

[10] J. Colliander, "This is fake news": Investigating the role of conformity to other users' views when commenting on and spreading disinformation in social media, Comput. Hum. Behav. 97 (2019) 202–215.

[11] H.T. Nguyen, T.N. Dinh, M.T. Thai, Revisiting of 'revisiting the stop-and-stare algorithms for influence maximization', in: Computational Data and Social Networks, Springer International Publishing, 2018, pp. 273–285.

[12] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003, pp. 137–146.

[13] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2007, pp. 420–429.

[14] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 199–207.

[15] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 1029–1038.

[16] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, K. Xie, Simulated annealing based influence maximization in social networks, in: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI Press, 2011, pp. 127–132.

[17] K. Jung, W. Heo, W. Chen, IRIE: Scalable and robust influence maximization in social networks, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 918–923.

[18] F. Kazemzadeh, A.A. Safaei, M. Mirzarezaee, S. Afsharian, H. Kosarirad, Determination of influential nodes based on the communities' structure to maximize influence in social networks, Neurocomputing 534 (2023) 18–28.

[19] Y. Yang, X. Mao, J. Pei, X. He, Continuous influence maximization: What discounts should we offer to social network users? in: Proceedings of the 2016 International Conference on Management of Data, ACM, 2016, pp. 727–741.

[20] N. Ohsaka, T. Sonobe, S. Fujita, K. ichi Kawarabayashi, Coarsening massive influence networks for scalable diffusion analysis, in: Proceedings of the 2017 ACM International Conference on Management of Data, ACM, 2017, pp. 635–650.
[21] G. Zhao, P. Jia, A. Zhou, B. Zhang, InfGCN: Identifying influential nodes in complex networks with graph convolutional networks, Neurocomputing 414 (2020) 18–26.
[22] J. Kou, P. Jia, J. Liu, J. Dai, H. Luo, Identify influential nodes in social networks with graph multi-head attention regression model, Neurocomputing 530 (2023) 23–36.
[23] S. Bharathi, D. Kempe, M. Salek, Competitive influence maximization in social networks, in: Proceedings of the 3rd International Conference on Internet and Network Economics, Springer-Verlag, 2007, pp. 306–311.
[24] S. Bhagat, A. Goyal, L.V. Lakshmanan, Maximizing product adoption in social networks, in: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, ACM, 2012, pp. 603–612.
[25] I. Valera, M. Gomez-Rodriguez, Modeling adoption and usage of competing products, in: 2015 IEEE International Conference on Data Mining, IEEE, 2015, pp. 409–418.
[26] H. Li, S.S. Bhowmick, J. Cui, Y. Gao, J. Ma, GetReal: Towards realistic selection of influence maximization strategies in competitive networks, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, ACM, 2015, pp. 1525–1537.
[27] Y. Zhu, D. Li, Z. Zhang, Minimum cost seed set for competitive social influence, in: IEEE INFOCOM 2016 - the 35th Annual IEEE International Conference on Computer Communications, IEEE, 2016, pp. 1–9.
[28] W. Hong, C. Qian, K. Tang, Efficient minimum cost seed selection with theoretical guarantees for competitive influence maximization, IEEE Trans. Cybern. 51 (12) (2021) 6091–6104.
[29] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2013, pp. 946–957.
[30] Y. Tang, X. Xiao, Y. Shi, Influence maximization: Near-optimal time complexity meets practical efficiency, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, ACM, 2014, pp. 75–86.
[31] Y. Tang, Y. Shi, X. Xiao, Influence maximization in near-linear time: A martingale approach, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, ACM, 2015, pp. 1539–1554.
[32] H.T. Nguyen, M.T. Thai, T.N. Dinh, Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks, in: Proceedings of the 2016 International Conference on Management of Data, ACM, 2016, pp. 695–710.
[33] K. Huang, S. Wang, G. Bevilacqua, X. Xiao, L.V.S. Lakshmanan, Revisiting the stop-and-stare algorithms for influence maximization, Proc. VLDB Endow. 10 (9) (2017) 913–924.
[34] X. Wang, Y. Zhang, W. Zhang, X. Lin, C. Chen, Bring order into the samples: A novel scalable method for influence maximization, IEEE Trans. Knowl. Data Eng. 29 (2) (2017) 243–256.
[35] Q. Guo, S. Wang, Z. Wei, W. Lin, J. Tang, Influence maximization revisited: Efficient sampling with bound tightened, ACM Trans. Database Syst. 47 (3) (2022) 1–45.
[36] Y. Zhu, J. Tang, X. Tang, S. Wang, A. Lim, 2-hop+ sampling: Efficient and effective influence estimation, IEEE Trans. Knowl. Data Eng. 35 (2) (2023) 1088–1103.
[37] D. Myers, J. Twenge, Social Psychology, McGraw-Hill Education, 2022.
[38] S.E. Asch, Opinions and social pressure, Sci. Am. 193 (5) (1955) 31–35.
[39] J.C. Coultas, E.J.C. van Leeuwen, Conformity: Definitions, types, and evolutionary grounding, in: Evolutionary Perspectives on Social Psychology, Springer, 2015, pp. 189–202.
[40] H. Li, S.S. Bhowmick, A. Sun, CASINO: Towards conformity-aware social influence analysis in online social networks, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, ACM, 2011, pp. 1007–1012.
[41] J. Zhang, J. Tang, H. Zhuang, C.W.-K. Leung, J. Li, Role-aware conformity influence modeling and analysis in social networks, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI Press, 2014, pp. 958–964.
[42] J. Tang, S. Wu, J. Sun, Confluence: Conformity influence in large social networks, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 347–355.
[43] H. Li, S.S. Bhowmick, A. Sun, J. Cui, Conformity-aware influence maximization in online social networks, The VLDB J. 24 (1) (2014) 117–141.
[44] Y. Li, X. Gan, L. Fu, X. Tian, Z. Qin, Y. Zhou, Conformity-aware influence maximization with user profiles, in: 2018 10th International Conference on Wireless Communications and Signal Processing, WCSP, IEEE, 2018, pp. 1–6.
[45] H. Li, H. Li, S.S. Bhowmick, CHASSIS: Conformity meets online information diffusion, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, ACM, 2020, pp. 1829–1840.
[46] W. Chen, Homepage of Wei Chen, https://www.microsoft.com/en-us/research/people/weic/selected-projects/.
[47] J. Leskovec, Stanford large network dataset collection, http://snap.stanford.edu/data/index.html.

Yonggang Liu is a Ph.D. candidate in computer science and technology from Hunan University, China. His research interests include social network analysis, parallel and distributed processing, and database systems.

Yikun Hu get the Ph.D. degree at Hunan University, China. His research interests are mainly in parallel and distributed processing, Cluster, Grid and Cloud computing.

Siyang Yu received the Ph.D. degrees in computer science and technology from Hunan University, China, in 2017. He is currently a teacher at Hunan University of Finance and Economics. His research interests include industrial internet of things, abnormal analysis, and intrusion detection.

Xu Zhou received the Ph.D. degree in computer science and technology from Hunan University, China, in 2016. She is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University. She has published over 30 papers in international journals and conferences, such as the IEEE Transactions on Knowledge and Data Engineering, the IEEE Transactions on Parallel and Distributed Systems, and the IEEE Transactions on Computers. Her current research interests include parallel and distributed processing, and database systems.

Keqin Li is a SUNY Distinguished Professor of computer science with the State University of New York. He is also a National Distinguished Professor with Hunan University, China. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber–physical systems, heterogeneous computing systems, big data computing, high-performance computing, CPU–GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent and soft computing. He has authored or coauthored more than 780 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He holds over 60 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top 10 most influential scientists in distributed computing based on a composite indicator of Scopus citation database. He has chaired many international conferences. He is currently an associate editor of the ACM Computing Surveys and the CCF Transactions on High Performance Computing. He has served on the editorial boards of the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Computers, the IEEE Transactions on Cloud Computing, the IEEE Transactions on Services Computing, and the IEEE Transactions on Sustainable Computing. He is an IEEE Fellow.