Risk Sciences 1 (2025) 100019

Contents lists available at ScienceDirect

Risk Sciences

journal homepage: www.keaipublishing.com/en/journals/risk-sciences/



Research article

Data-driven decision making for social influence risk management☆

Siqi Wang, Jiahao Xie, Yifan Wang*, Guangmo Tong

University of Delaware, 210 S College Ave, Newark, DE 19716, USA

ARTICLE INFO

Keywords:
Data-driven optimization
Social network
Social influence risk management
Structure support vector machine

ABSTRACT

Social influence risk management tasks, such as misinformation prevention and source detection, have gained increasing attention due to the rapid development of social networks, which play an important role in managing the spread of both positive and harmful information. Previous studies for solving these tasks mainly focus on developing exact or approximate approaches where comprehensive diffusion model information is provided. However, it is usually challenging to obtain such information in practice. In this paper, we propose SocialLearner, a novel data-driven decision-making framework designed for social influence risk management tasks in scenarios where diffusion model information is not fully accessible. This approach employs a two-step optimization process that integrates model ensemble techniques with an optimized sampling distribution strategy, aiming to enhance the accuracy of approximating unknown objective functions by leveraging historical pairs of inputs and decisions. Furthermore, extensive experiments on various social influence risk management tasks and network structures are conducted to evaluate the performance of SocialLearner. The results demonstrate its superior accuracy and scalability compared to state-of-the-art methods, highlighting its robustness as a solution in addressing social influence risk management tasks.

1. Introduction

The rapid development of social networks has connected people around the world, allowing efficient information exchange that facilitates individual communication, knowledge sharing, and brainstorming (Bakshy et al., 2012; Guille et al., 2013; Li et al., 2017). However, these benefits have also raised concerns due to their role in supporting the spread of misinformation, which causes significant damage at individual, societal, and even global levels (Nguyen et al., 2013; Zhang et al., 2016; Zareie & Sakellariou, 2021). Its consequences include affecting public trust (Ognyanova et al., 2020), amplifying societal polarization (Azzimonti & Fernandes, 2023), and influencing decisions in critical domains such as healthcare, politics, and public safety (Muhammed T & Mathew, 2022). Social influence risk management (Nguyen et al., 2012; Islam et al., 2020; Tong, 2022) is an important field focusing on handling risks that arise from the diffusion of misinformation within social networks. Existing research mainly focuses on three key tasks:

Peer review under the responsibility of the Editorial Board of Risk Sciences.

☆ This work is an extended version of the conference paper presented at CSoNet 2024. The current version expands the MetaLearner framework to address the social influence risk management problems with new experimental results and analysis of these tasks.

* Corresponding author.
E-mail addresses: wsqbit@udel.edu (S. Wang), jiahaox@udel.edu (J. Xie), yifanw@udel.edu (Y. Wang), amotong@udel.edu (G. Tong).

https://doi.org/10.1016/j.risk.2025.100019
Received 27 December 2024; Received in revised form 2 April 2025; Accepted 15 April 2025
Available online 22 April 2025
2950-6298/© 2025 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

misinformation prevention (Nguyen et al., 2013; Farajtabar et al., 2017; Lewandowsky & Van Der Linden, 2021), sources of misinformation detection (Nguyen et al., 2012; Shelke & Attar, 2019; Torabi Asr & Taboada, 2019), and communities protection from misinformation influence (Rice et al., 2021; Budak et al., 2011).

Previous research on social influence risk management tasks has primarily concentrated on designing exact and approximate algorithms (Zhu & Ying, 2014; Shelke & Attar, 2019; Tong & Du, 2019). These algorithms aim to identify optimal solutions by utilizing the structural information of networks and underlying diffusion models. However, real-world social networks are vast, noisy, and frequently evolving (Freeman, 2004; Tabassum et al., 2018), making it challenging to obtain a complete and accurate underlying diffusion model. Additionally, the inherent stochasticity in the information spreading process makes it even more difficult to accurately model and predict the impact of misinformation. These challenges pose new requirements for designing more robust methods for social influence risk management that can perform effectively with limited social network information.

Limited access to complete network and diffusion model data results in an unknown objective function for solving social influence risk management tasks, where traditional methods cannot be applied directly. To address this challenge, Tong proposes StratLearner (Tong, 2020), which approximates the objective function using randomly sampled subgraphs as well as historical input-decision pairs for the misinformation prevention problem. To deal with broader data-driven decision-making tasks, Tong further generalizes such strategies into the USCO-solver (Tong, 2021) framework. However, the USCO-solver faces two limitations: 1) Its performance heavily relies on the number of sampled subgraphs. While increasing subgraphs improves accuracy, it also increases training time and memory usage; 2) The USCO-solver leads to diminishing returns. The performance gains are limited even with larger training datasets. Consequently, it often converges to suboptimal solutions similar to those observed in (Wortsman et al., 2022).

To address the above limitations, we propose SocialLearner, a universal two-step framework designed for solving social influence risk management tasks in scenarios with restricted access to diffusion model information. In the first step, inspired by the ensemble model and representation learning, we develop a novel subgraph sampling method using historical input-decision pairs. Specifically, multiple models are trained on various sub-training datasets and sampled subgraphs to estimate the objective function, and we treat the results of each model as a distinct representation of the social network graph. Ideally, deploying a large number of such trained models would replace the random subgraph sampling process, thereby significantly improving estimation accuracy. However, computational constraints make such a method impractical. To overcome this limitation, in the second step, we introduce a distribution estimation strategy that decreases the number of required models while still generating high-quality subgraphs for accurate objective function estimation. In the end, once the objective function is estimated with sufficient accuracy, traditional methods that are designed for the target social influence risk management task can be employed to optimize the objective function for new inputs.

Overall, our contributions are summarized: 1) We propose SocialLearner, a novel universal framework inspired by model ensemble techniques, which employs a two-step learning process to enhance the accuracy of objective function estimation; 2) We introduce a distribution estimation strategy that reduces the number of required models to a quantity determined by hyperparameter. In practice, this approach can scale down the number of required models to as low as 1 % or even 0.1 % of the original amount, thereby significantly decreasing the computational and memory costs for real-world applications; 3) Extensive experiments conducted on three tasks show that SocialLearner consistently outperforms state-of-the-art methods in terms of accuracy across various social network structures, demonstrating its robustness and scalability for large-scale social influence risk management tasks.

2. Related work

The social influence of risk management has been widely studied within the social network community. In this paper, we focus on three tasks, including misinformation prevention, source detection, and community protection, which are acknowledged as representative problems in this field (Budak et al., 2011; Islam et al., 2020).

Misinformation prevention concentrates on selecting a set of seed nodes to spread truthful information as a countermeasure against misinformation. This task was first formally proposed by (Budak et al., 2011), who introduced the concept of competing campaigns and formulated the problem using the multi-campaign independent cascade model. Their work motivates significant research interest, leading to the development of greedy algorithms for approximate solutions under various diffusion models (Fan et al., 2013; He et al., 2012). Another line of research focused on developing heuristic algorithms for misinformation prevention (Ping et al., 2014; Zhang et al., 2015). In addition, there are also some machine learning based methods trying to solve misinformation prevention over billion-sized graphs (Manchanda et al., 2020). A comprehensive survey of such work can be found in (Islam et al., 2020; Zareie & Sakellariou, 2021; Su et al., 2023).

Source detection aims to identify the origin of information spread according to partial observations of the final states of influenced nodes. This research area gained attention starting with the Susceptible-Infected (SI) model (Shah & Zaman, 2010,2011,2012). Based on this foundation, heuristic algorithms for source detection under the Susceptible-Infected-Recovered (SIR) model were introduced in (Zhu & Ying, 2014). Another line of research focused on applying machine learning methods for source detection (Dong et al., 2019; Cheng et al., 2024). Further studies can refer to (Shelke & Attar, 2019; Su et al., 2023; Zhang et al., 2024).

Community protection seeks to disrupt connections within a community to safeguard it from hostile influences and can be simplified as a minimum cut problem—a well-known combinatorial optimization challenge (Stoer & Wagner, 1994; Karger & Stein, 1996).

Most existing approaches rely on prior knowledge of the diffusion model, but it may not be readily available in real-world scenarios. To address this limitation, (Tong, 2020) proposed the StratLearner framework, which directly learns to predict solutions from historical query-decision pairs, making it suitable for scenarios where diffusion models are unknown. Building on this, Tong introduced the USCO-solver (Tong, 2021), a more general framework for data-driven decision-making problems. However, both StratLearner and USCO-solver suffer from high computational overhead and limited accuracy. Our proposed SocialLearner is capable

of dealing with these by combining model ensemble techniques with an optimized sampling distribution strategy, thus achieving better performance.

3. Preliminary

In this section, we introduce the considered diffusion model and formally define the social influence risk management tasks, as well as provide a formal statement of our research problem.

3.1. Diffusion model

A social network can be modeled as a directed graph $G = (V, E)$, where V is the set of users and E is the edge set. The existence of each edge $e = (u, v) \in E$ indicates that user $u \in V$ can influence user $v \in V$. Let $N(u) \subseteq V$ be the neighbors of node u. Regarding information diffusion, suppose that $C \in \mathbb{Z}^+$ cascades are diffused in the social network. During the diffusion process, each node $v \in V$ can be c_i-active with $i \in [C]$ or inactive. A diffusion model M_G is defined on a directed graph G, where each edge $e = (u, v)$ is associated with two parameters $p(e) \in [0, 1]$ and $t(e) \in \mathbb{R}^+$. $p(e)$ represents the probability of user u influence user v, and $t(e)$ is a random variable sampled from a distribution $\mathscr{T}_e$, indicating the time required for u to influence v. To this end, the diffusion process can be defined as follows.

- **Initialization:** Each cascade c_i randomly selects its seeds $S_i \subseteq V$ with $i \in [C]$. Specifically, for all $i \neq j$, we have $S_i \cap S_j = \emptyset$. Each edge e samples a weight $t(e)$ following $\mathscr{T}_e$.
- **Time 0:** The nodes within S_i are c_i-active, while the remaining nodes are inactive.
- **Time step T**: Each node u that is newly activated by c_i attempts to influence its inactive neighbors $v \in N(u)$ with a success probability of $p((u, v))$. If successful, node v will become c_i-active and start to influence its inactive neighbors at time step $T + t(u, v)$. In the case of a tie, where a node is successfully activated by multiple cascades, the node adopts the cascade with the smallest index.
- **Termination:** The diffusion process terminates when no more inactive nodes can be activated.

3.2. Social influence risk management

Given the diffusion model described above and the initial activation status of the underlying social network, the social influence risk management tasks often involve determining the optimal intervention strategy to minimize the spread of misinformation or maximize the target influence. Note that the proposed diffusion model is highly expressive due to the flexibility in designing $\mathscr{T}_e$, which subsumes several classic diffusion models (Kempe et al., 2003).

Formally, let $x \in \mathscr{X}$ represent a specific input, such as a target intervention or detection scenario, where $\mathscr{X}$ denotes the space of all possible inputs. A feasible solution to x, referred to as a decision $y \in \mathscr{Y}$, represents an intervention strategy, such as selecting nodes to inoculate against misinformation or identifying key influences, where we use $\mathscr{Y}$ to denote the decision space. Given a diffusion model M_G, an input x, and a decision y, we use $F(x, y, M_G)$ to evaluate the quality of the decision y.

$$F(x, y, M_G) = \int_{g \in \mathscr{G}} f(x, y, g) \cdot \Phi(g) d\Phi(g), \tag{1}$$

where a) $f(x, y, g) \in \mathbb{R}$ is a function that evaluates the quality of the decision y on a specific weighted subgraph g of G, representing one possible state of the network. In particular, we denote the edge set of g as E_g, where the weight $t(e)$ of each edge $e \in E_g$ is sampled following $\mathscr{T}_e$; b) $\mathscr{G}$ denotes the space of all such possible weighted subgraphs; and c) $\Phi(g) \in [0, 1]$ indicates the probability of sampling subgraph g from $\mathscr{G}$ and can be calculated as:

$$\Phi(g) = \prod_{e \in E_g} (p(e) \cdot h_{\mathscr{T}_e}(t(e))) \times \prod \prod_{e \in E \setminus E_g} (1 - p(e)), \tag{2}$$

where $h_{\mathscr{T}_e}(t(e))$ is the probability density function of the distribution $\mathscr{T}_e$ evaluated at $t(e)$. The objective of the influence risk management task is to obtain a decision y^* such that the following function is maximized, where

$$y^* = \arg\max_{y \in \mathscr{Y}} F(x, y, M_G). \tag{3}$$

In this paper, we focus on three popular tasks in social influence risk management: Misinformation Prevention (MP), Source Detection (SD), and Community Protection (CP). We give the formal problem statements for these three tasks as follows.

MP. Given a diffusion model M_G, a node set $S_0 \subseteq V$, and a budget $r \in \mathbb{Z}^+$, the objective of MP is to select a set of nodes $S_1 \subseteq V$ with $|S_1| \leq r$ and $S_1 \cap S_0 = \emptyset$ such that the number of nodes that are not c_0-active after the diffusion process can be maximized. Let $F_1(S_0, S_1, M_G)$ represent the number of nodes that are not c_0-active at the end of the diffusion process. The objective function $F(S_0, S_1, M_G)$ is defined as follows.

$$F(S_0, S_1, M_G) = F_1(S_0, S_1, M_G) - F_1(S_0, \emptyset, M_G)$$

where $F_1(S_0, \emptyset, M_G)$ represents the number of nodes that are not C_0-active when no positive nodes are initially selected. Formally, the misinformation prevention problem aims to find a node set S_1 such that the objective function is maximized.

$$\arg \max_{S_1 \subseteq V, S_1 \cap S_0 = \emptyset, |S_1| \leq r} F(S_0, S_1, M_G).$$

SD. Given a diffusion model M_G and two node sets $V_0, V_1 \subseteq V$, the objective of SD is to select a node $v \in V$ as S_0 to maximize the probability that nodes in V_0 are c_0-active and nodes in V_1 are not c_0-active after the diffusion process. The objective function $F((V_0, V_1), v, M_G) \in [0, 1]$ represents the probability. Formally, the source detection problem aims to find a node v such that the objective function is maximized.

$$\arg \max_{v \in V} F((V_0, V_1), v, M_G).$$

CP. Given a diffusion model M_g and two nodes $v_0, v_1 \in V$, the objective of CP is to select an edge set $E' \subseteq E$ to remove and let v_0 and v_1 can not influence each other. We want to minimize the total time cost of the edge set E'. The objective function $F((v_0, v_1)), E', G)$ is defined as follows.

$$F((v_0, v_1), E', M_G) = \sum\nolimits_{e \in E'} \mathbb{E}(t(e)) \cdot p(e).$$

Formally, the community protection problem aims to find an edge set E^*such that the objective function is maximized.

$$E^* = \arg \max_{E' \subseteq E} F((v_0, v_1), E', M_G).$$

3.3. Problem statement

For social influence risk management, to obtain a decision, we need to solve equation (3). Currently, there exist reliable solvers for different social influence risk management tasks when all information about M_G is known. However, the main challenge lies in the inaccessibility of distribution $\mathscr{T}_e$. To address this challenge, we propose SocialLearner, a learning framework that can approximate the objective function by leveraging historical input-decision pairs. Specifically, the training dataset is defined as $D = \{(x_i, y_i)\}_{i \in [C]}$, which includes the pairs of previously observed inputs and their corresponding decisions. Suppose that each pair of the dataset follows the distribution $\mathscr{D}$. After obtaining the objective function, one can use the existing reliable solvers as an oracle to obtain a decision y. In general, SocialLearner serves as a predictive model $\mathscr{A}$ that can generate high-quality decisions $\hat{y} = \mathscr{A}_{\mathscr{D}}(x)$ for unseen inputs $x \in \mathscr{X}$. The quality of the generated solution is evaluated by comparing it with the optimal solution y using the following ratio.

$$\frac{F(x, \hat{y}, M_G)}{F(x, y, M_G)}, \tag{4}$$

where a higher ratio indicates a closer approximation to the optimal solution. Thus, the learning problem is framed as minimizing the following loss function:

$$\mathscr{L} = \mathbb{E}_{(x,y) \sim \mathscr{D}} \left(1 - \frac{F(x, \hat{y}, M_G)}{F(x, y, M_G)} \right), \tag{5}$$

which penalizes the framework $\mathscr{A}_{\mathscr{D}}$ for generating solutions that deviate from the optimal in terms of the objective function F.

4. Method

In this section, we present SocialLearner, a framework designed to address the challenge of social influence risk management when the diffusion model M_G cannot be fully accessed. SocialLearner aims to estimate the objective function $F(x, y, M_G)$ by leveraging historical input-decision data. Our framework consists of two main components: optimization of the sampling method and refinement of the objective function. We first introduce the overall framework.

4.1. Overall approach

Following the idea proposed in (Tong, 2020), the problem of social influence risk management can be addressed by sampling weighted subgraphs from a distribution ϕ and approximating the objective function as follows.

$$F(x, y, M_G) = \mathbf{W}^T \cdot \mathbf{S}, \quad \mathbf{W}, \mathbf{S} \in \mathbb{R}^n, \tag{6}$$

where $\mathbf{S} = [s_1, s_2, \cdots, s_n]^T$ and $s_i = f(x, y, g_i)$ represents the objective value computed on the i-th sampled weighted graph g_i. The vector $\mathbf{W} = [w_1, w_2, \cdots, w_n]^T$, where w_i contains the weights associated with the corresponding objective values.

Previous research highlights that the accuracy of the objective function approximation heavily depends on the quality of the generated weighted subgraphs. A randomly chosen distribution ϕ about sampling a subgraph g from $\mathscr{G}$ often results in suboptimal performance. To address this limitation, the SocialLearner employs a two-step structure. In the sampling method optimization step, we first initialize a graph sampling distribution ϕ. The SocialLearner trains N different models, where each of them uses n different subsets of training data D and K subgraphs sampled from ϕ, to update the sampling distribution and generate a better sampling distribution $\phi^{'}$. This step seeks to enhance the quality of the generated subgraphs by adapting the distribution to the specific problem context of social influence risk management tasks. In the second step, using the learned sampling distribution $\phi^{'}$, SocialLearner refines the approximation of the objective function, leveraging the improved subgraph quality to enhance performance in the objective function optimization step.

Algorithm 1. SocialLearner

1: **Input:** $N, n, K \in \mathbb{Z}^+, \phi, \mathcal{D}$;
2: Update the sampling distribution ϕ to an optimized distribution ϕ' using Algorithm 2;
3: Estimate the objective function via Algorithm 3;
4: **Return:** an estimate of the objective function $F(x, y, G)$;

The overall structure of the SocialLearner is presented in Algorithm 1. Detailed explanations of the sampling method optimization and objective function optimization are provided in the following subsections.

4.2. Sampling method optimization

The motivation for sampling method optimization stems from the concept of multi-head attention introduced in (Vaswani et al., 2017). Similar to how multi-head attention captures different aspects of input data, the SocialLearner aims to capture different aspects of social network diffusion by leveraging diverse weighted subgraphs sampled from ϕ and distinct training datasets D. Specifically, to learn an optimized sampling distribution, we sample $K \in \mathbf{Z}^+$ weighted subgraphs $\mathbf{g} = \{g_1, \cdots, g_K\}$ using ϕ and sample n input-decision pairs from $\mathscr{D}$.

To optimize the sampling distribution, a Structured Support Vector Machine (SSVM) (Tsochantaridis et al., 2005) is trained using the cutting-plane algorithm (Joachims et al., 2009). The optimization problem is formulated as follows:

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{W}\|_2^2 + \frac{C}{2n}\sum_{i=1}^{n}\xi_i \\ &\text{s.t.} \sum_{k=1}^{K} w_k{\cdot}f(x_i, y_i, g_k) - \sum_{k=1}^{K} w_k{\cdot}f(x_i, y^*, g_k) \geq \alpha{\cdot}H(y_i, y^*) - \xi_i, \\ &\forall\, i \in [n],\ \forall\, (x_i, y_i) \in D^j,\ \forall\, y^*{:}\ y^* \in \mathscr{Y},\ y^* \neq y_i, \end{aligned} \tag{7}$$

where $C, \alpha \in \mathbb{R}^+$ are hyperparameters. The function $H(y_i, y^*)$ measures the similarity between two intervention strategies. Since there may exist exponential constraints in Equation (7), we need an Oracle that can efficiently find the solution of:

$$\arg\max \sum_{k=1}^{K} w_k{\cdot}f(x_i, y^*, g_k) \tag{8}$$

Based on the optimized sampling distribution, the SocialLearner estimates a static graph as follows:

$$t(e) = \sum_{i=1}^{K} w_i{\cdot}t_i(e),$$

where $t_i(e)$ represents the diffusion time of edge e in subgraph g_i. If edge e is absent from subgraph g_i^j, then $t_i(e) = 0$. Using the static graph, the SocialLearner estimates the sampling distribution $\phi^{'}$ as a combination of Weibull distribution $\phi(e)$. For each edge e in the static graph, the scale parameter a is sampled from a range $[A, B]$ where $A, B \in \mathbf{Z}^+$. The shape parameter b is calculated by solving $\mathbb{E}[\text{Weibull}(a, b)] = t_i(e)$. Thus, $\phi(e) = \text{Weibull}(a, b)$, and the optimized sampling distribution is represented as $\phi^{'} = \{\phi(e)\}, e \in E$.

Algorithm 2. Sampling method optimization

Input: $N, n, K \in \mathbb{Z}^+, \phi, \mathcal{D}$;
for $j = 1, 2, \cdots, N$ **do**:
  Sample K subgraphs with the sampling method ϕ to construct $\mathbf{g}^j = \{g_1^j, \cdots, g_K^j\}, g_i^j = G(V_i^j, E_i^j, T_i^j)$;
  Sample n training pairs from the dataset $\mathcal{D}$ to construct D^j;
  Calculate $\mathbf{W}^j = \{w_1^j, \cdots, w_K^j\}$ for $D^j, \mathbf{g}^j$ by applying cutting plane algorithm [44] to solve Equation 7;
  Construct an edge set $E^j = \bigcup_{i=1}^K E_i^j$, where E_i^j is the edge set of subgraph g_i^j;
  for $e \in E_i^j$ **do**:
    Calculate the weighted weight $T^j(e) = \sum_{i=1}^K w_i^j \cdot t_i^j(e)$, $t_i^j(e) = 0$ if $e \notin g_i^j$;
    Sample a from a uniform distribution $[A, B]$, with positive integers A, B;
    Calculate b by solving $\mathbb{E}[\text{Weibull}(a, b)] = t^j(e)$;
    $\phi'_j(e) = \text{Weibull}(a, b)$;
  end for
  $\phi'_j = \{\phi'_j(e)\}, e \in E^j$;
end for
Return a set of sampling methods $\phi' = \{\phi'_1, \cdots, \phi'_N\}$;

The SocialLearner derives *N* distinct sampling distributions, and the overall process is summarized in Algorithm 2.

4.3. Objective function optimization

After obtaining the sampling distribution set $\{\phi_1', \cdots, \phi_N'\}$, as shown in Algorithm 3, SocialLearner samples $\frac{K}{N}$ subgraphs from each optimized sampling distribution. This results in a new subgraph set $\mathbf{g}'$. With a new training dataset D' sampled from $\mathscr{D}$, we can train additional SSVM based on $\mathbf{g}'$ and D' to optimize the objective function, defined as:

$$F(x, y, G) = \sum_{i=1}^K w_i' f(x, y, g_i'), w_i' \in \mathbf{W}', g_i' \in \mathbf{g}' \tag{9}$$

Algorithm 3. Objective function optimization

1: Input: $n, K \in \mathbb{Z}^+$, $\{\phi^1, \cdots, \phi^N\}$, $\mathcal{D}$;
2: Sample $\frac{K}{N}$ subgraphs from each new sampling method and combine them as a new group of subgraphs $\mathbf{g}'$;
3: Sample n training data from $\mathcal{D}$ as training dataset D';
4: Calculate $\mathbf{W}' = \{w'_1, \cdots, w'_K\}$ for $D', \mathbf{g}'$ by applying cutting plane algorithm [44] to solve problem 7;
5: Return the final estimation of objective function as $F(x, y, G) = \sum_{i=1}^{K} w'_i \cdot f(x, y, g'_i), w'_i \in \mathbf{W}', g'_i \in \mathbf{g}'$;

It is important to note that sampling from the optimized distributions is constrained by computational resources. If sufficient computational resources were available, a preferable approach would involve training $N = K$ models during the sampling optimization step and directly using the group of static graphs as $\mathbf{g}'$ for further processing. However, this approach becomes impractical due to the significant computational and time resources required. To address this challenge, we leverage distribution estimation to approximate the learned static graphs as a distribution. This approximation maintains a balance between the effectiveness and efficiency of SocialLearner, enabling scalable and practical implementation without excessive computational overhead.

5. Experiment

In this section, we present empirical studies on our proposed method for different social risk tasks across various graph datasets and evaluate the performance of the SocialLearner with two baselines, USCO Solver and Strat Learner.

5.1. Graph dataset

Our approach is evaluated on four graph datasets that are generated from classic graph models, i.e., Erdös-Rényi graph (ER) (Erdös and Rényi, 1959), power-law graph (Pow), Colorado road network graph (Col) (Demetrescu et al., 2009), and Kronecker graph (Kro) (Leskovec et al., 2010). Regarding the diffusion model M_G, for each generated graph G, we assume that each edge weight $t(e)$ follows a Weibull distribution $\mathscr{T}_e = \text{Weibull}(a, b)$ with parameters a and b sampled from the set $\{1, \cdots, 10\}$ independently, and the weight set to the expected value of this distribution. The edge probability $p(e)$ is set as $1/\mathscr{N}(v)$, $\mathscr{N}(v)$ is the number of neighborhood of v for edge (u, v). The graph information is shown in Table 1.

5.2. Misinformation prevention problem

Sample generation. We generate a dataset with 10,000 input-decision pairs. For each input x, we sample an integer $|C_1|$ from a power-law distribution with a shape parameter of 2.5 and a scale of 10. We then randomly select $|C_1|$ nodes from the set V to form C_1. We let r equals $|C_1|$. We select the HMP algorithm (Tong & Du, 2019) as oracle and for each input x, we use it to find the corresponding decision $y \in \mathscr{Y}$.

Experiment settings. The training size n' for experiment is selected from a set {270, 540, 1080, 2160} and the testing size is 270. For representing the results, we report the ratio $\frac{F(x,\hat{y},M_G)}{F(x,y,M_G)}$, where $\hat{y}$ is the decision made by StratLearner (Tong, 2020). Each $f(x, y, G)$ is calculated through 10000 simulations. The number of weighted graphs K' is set to {100, 400, 800, 1600}. Specifically, for each model trained in step one, we set $N = 10$ and use $n = \{27, 54, 108, 216\}$ samples for training each model and 270 samples for testing, also reporting the ratio $\frac{f(x,\hat{y},G)}{f(x,y,G)}$. The number of weighted graphs K is set to {10, 40, 80, 160}. This ensures that the total number of training

Table 1
Statistics on the graph data.

Dataset	# node	# edge	Reference
ER	512	638	(Erdös and Rényi, 1959)
Pow	768	1532	(Adamic et al., 2001)
kro	1024	2655	(Leskovec et al., 2010)
col	511	1551	(Demetrescu et al., 2009)

Table 2
Results for the Misinformation Prevention on Kro Graph. Each cell shows the average ratio compared with the sample decision, together with the standard deviation (std). The total number of sampled graphs K' are 100, 400, 800, and 1600, respectively.

	Training	Number of sampled grpah			
	Size n'	100	400	800	1600
Strat-Learner	270	0.699 (8E−3)	0.759 (7E−3)	0.785 (1E−2)	0.810 (9E−3)
	540	0.707 (5E−3)	0.743 (8E−3)	0.780 (9E−3)	0.813 (7E−3)
	1080	0.708 (2E−2)	0.760 (1E−2)	0.782 (8E−3)	0.817 (5E−3)
	2160	0.701 (1E−2)	0.756 (1E−2)	0.792 (5E−3)	0.821 (8E−3)
Social-Learner	270	0.698 (5E−3)	0.777 (8E−3)	0.807 (1E−2)	0.819 (9E−2)
	540	0.694 (1E−2)	0.772 (8E−3)	0.805 (5E−3)	0.827 (8E−2)
	1080	0.704 (8E−3)	0.773 (6E−3)	0.807 (9E−3)	0.831 (8E−2)
	2160	0.710 (1E−2)	0.773 (1E−2)	0.811 (9E−3)	0.834 (9E−2)
Other Method		**Random**	0.190 (5E−3)		

samples and weighted graphs remain consistent across experiments, such that $N \times n = n'$ and $N \times K = K'$. For each (n, K) pair, we repeat the experiment five times and report the mean ratio and corresponding variance. The hyperparameter α in Equation (7) is set to 1000, and C is set to 0.01. In the StratLearner, each subgraph is generated by selecting each edge independently at random with a probability of 0.01 and assigning a fixed weight of 1.0 to each selected edge. We denote such distribution as $\phi_{1.0}^{0.01}$. In the SocialLearner approach, subgraphs are also generated with a 0.01 probability for selecting each edge, but the edge weights are sampled from a different Weibull(c, d) distribution, with c and d independently sampled from $\{1, \cdots, 10\}$ for each edge during each sampling process. The results for the misinformation prevention problem are reported in Tables 2,3, and 4. A random algorithm that randomly selects a seed set C_1 is implemented as a baseline.

5.3. Source detection problem

Sample generation. We generate 10,000 input-decision pairs. For the input x, we sample a random node from V as the initial source S_0 and simulate a single diffusion process starting from S_0. From the resulting set of C_0-active nodes, we randomly select 10 nodes to form V_1, and from the remaining inactive nodes, we select 10 nodes to form V_2. If fewer than 10 nodes are available for either V_1 or V_2, we select as many as possible. For the Oracle, we use Monte Carlo sampling with 10,000 simulation runs to estimate $F((V_0, V_1), v, G)$ for each $v \in V$ to estimate the probability that the nodes in V_1 are C_0-active while those in V_2 are inactive. The nodes with the highest probability under such simulation are selected as the optimal decision y.

Experiment settings. The experiment settings are the same as those in the experiment on the misinformation prevention problem. The experiment results are shown in Table 5. We also implement a random algorithm that randomly selects a seed set C_0 as a baseline.

5.4. Community protection

Sample generation. To construct a dataset for the community protection problem, we enumerate all possible node pairs (s,t) as inputs. For each input x, we use the Edmonds-Karp algorithm (Zadeh, 1972) to find the corresponding decision $y \in \mathscr{Y}$.

Experiment settings. Since we want to ensure that v_0 and v_1 can not influence each other, sampling subgraphs is not appropriate in this experiment. Therefore, we sample weighed graphs instead. We set $n' = n = \{80, 160, 320\}$ samples for training each model and 960 samples for testing, reporting the ratio $\frac{f(x,y,G)}{f(x,\hat{y},G)}$. The number of weighted graphs K and K' is set to {40, 160, 640, 1600}. The experiment results are shown in Table 6.

Table 3
Results for the Misinformation Prevention on ER Graph. Each cell shows the average ratio compared with the sample decision, together with the standard deviation (std). The total number of sampled graphs K' is 100, 400, 800, and 1600, respectively.

	Training	Number of sampled grpah			
	Size n'	100	400	800	1600
Strat-Learner	270	0.661 (2E−2)	0.853 (6E−3)	0.873 (1E−2)	0.892 (3E−3)
	540	0.673 (2E−2)	0.861 (1E−2)	0.876 (6E−3)	0.897 (1E−2)
	1080	0.688 (3E−2)	0.844 (9E−3)	0.870 (1E−2)	0.899 (8E−3)
	2160	0.674 (2E−2)	0.857 (2E−2)	0.873 (5E−3)	0.903 (3E−3)
Social-Learner	270	0.678 (2E−2)	0.853 (2E−2)	0.913 (7E−4)	0.941 (3E−3)
	540	0.645 (2E−2)	0.862 (6E−3)	0.917 (1E−3)	0.938 (5E−3)
	1080	0.669 (3E−2)	0.858 (2E−2)	0.913 (1E−2)	0.936 (3E−3)
	2160	0.660 (3E−2)	0.868 (1E−2)	0.910 (6E−3)	0.937 (4E−3)
Other Method		**Random**	0.052 (2E−2)		

Table 4
Results for the Misinformation Prevention on Pow Graph. Each cell shows the average ratio compared with the sample decision, together with the standard deviation (std). The total number of sampled graphs $K^{'}$ is 100, 400, 800, and 1600, respectively.

	Training	Number of sampled grpah			
	Size $n^{'}$	100	400	800	1600
Strat-Learner	270	0.707 (1E−2)	0.839 (6E−3)	0.881 (1E−2)	0.902 (8E−3)
	540	0.686 (2E−2)	0.858 (8E−3)	0.878 (2E−2)	0.909 (9E−3)
	1080	0.680 (4E−2)	0.823 (2E−2)	0.890 (4E−3)	0.920 (7E−3)
	2160	0.682 (1E−2)	0.853 (2E−2)	0.889 (1E−2)	0.911 (3E−3)
Social-Learner	270	0.694 (2E−2)	0.936 (8E−3)	0.976 (3E−3)	0.976 (1E−3)
	540	0.697 (2E−2)	0.936 (1E−2)	0.978 (2E−3)	0.976 (1E−3)
	1080	0.688 (2E−2)	0.930 (4E−3)	0.978 (1E−3)	0.978 (1E−3)
	2160	0.710 (2E−2)	0.939 (6E−3)	0.978 (1E−3)	0.977 (2E−3)
Other Method		**Random**	0.047 (4E−3)		

Table 5
Results for the source detection on ER Graph. Each cell shows the average ratio compared with the sample decision, together with the standard deviation (std). The total number of sampled graphs $K^{'}$ is 100, 400, 800, and 1600, respectively.

	Training	Number of sampled grpah			
	Size $n^{'}$	100	400	800	1600
USCO-solver	270	0.570 (1E−2)	0.665 (9E−3)	0.682 (1E−2)	0.108 (2E−2)
	540	0.585 (2E−2)	0.685 (2E−2)	0.664 (2E−2)	0.716 (2E−2)
	1080	0.584 (5E−3)	0.688 (1E−2)	0.700 (2E−2)	0.707 (9E−3)
	2160	0.590 (1E−2)	0.595 (1E−2)	0.596 (1E−2)	0.859 (1E−2)
Social-Learner	270	0.563 (1E−2)	0.756 (1E−2)	0.859 (1E−2)	0.928 (2E−2)
	540	0.569 (2E−2)	0.763 (1E−2)	0.860 (1E−2)	0.945 (5E−3)
	1080	0.543 (3E−2)	0.775 (6E−2)	0.878 (1E−2)	0.985 (9E−3)
	2160	0.565 (2E−2)	0.761 (2E−2)	0.868 (1E−2)	0.931 (6E−2)
Other Method		**Random**	0.0003 (0)		

Table 6
Results for the community protection on Col Graph. Each cell shows the average ratio compared with the sample decision, together with the standard deviation (std).

	Training	Number of sampled grpah			
	Size $n^{'}$	40	160	640	1600
USCO-solver	80	0.235 (1E−2)	0.293 (2E−2)	0.341 (6E−2)	0.411 (1E−2)
	160	0.229 (3E−2)	0.280 (3E−2)	0.326 (3E−2)	0.438 (2E−2)
	320	0.215 (1E−2)	0.288 (2E−2)	0.331 (6E−2)	0.429 (1E−2)
Social-Learner	80	0.396 (8E−2)	0.280 (2E−2)	0.411 (2E−2)	0.488 (3E−2)
	160	0.364 (8E−2)	0.357 (3E−2)	0.429 (1E−2)	0.527 (1E−2)
	320	0.368 (8E−2)	0.355 (6E−2)	0.467 (1E−2)	0.548 (2E−2)
Other Method		**Random**	0.0002 (0)		

5.5. Analysis

Our main observation is that SocialLearner performs consistently better than USCO-solver, across various graph structures and task settings, when given the same amount of sampled graphs and sufficient training data. This observation indicates that SocialLearner effectively utilizes historical information to learn more reliable objective functions than the USCO-solver. Additionally, these results also highlight SocialLearner's stability in solving different data-driven decision-making problems. The robustness of SocialLearner makes it a better choice for data-driven decision making tasks in the social Influence risk management area.

Secondly, we observe that both SocialLearner and USCO-solver rely more heavily on the amount of the sampled graph rather than the amount of training data. This is an important observation because the availability of training data is frequently constrained due to cost or privacy considerations in real-world applications. We may have limited training data. However, since weighted graphs can be randomly sampled, we have the potential to get a sufficiently large number of graph samples. The reliance on graph sampling instead of extensive training data provides a practical advantage, making these methods applicable in scenarios where limited labeled data is available but graph structure can be observed or approximated.

We also conduct experiments to test the performance of StratLearner and SocialLearner without training on the ER graph for the misinformation prevention problem. The results are displayed in Fig. 1. It is clear that when the number of sampled graphs is

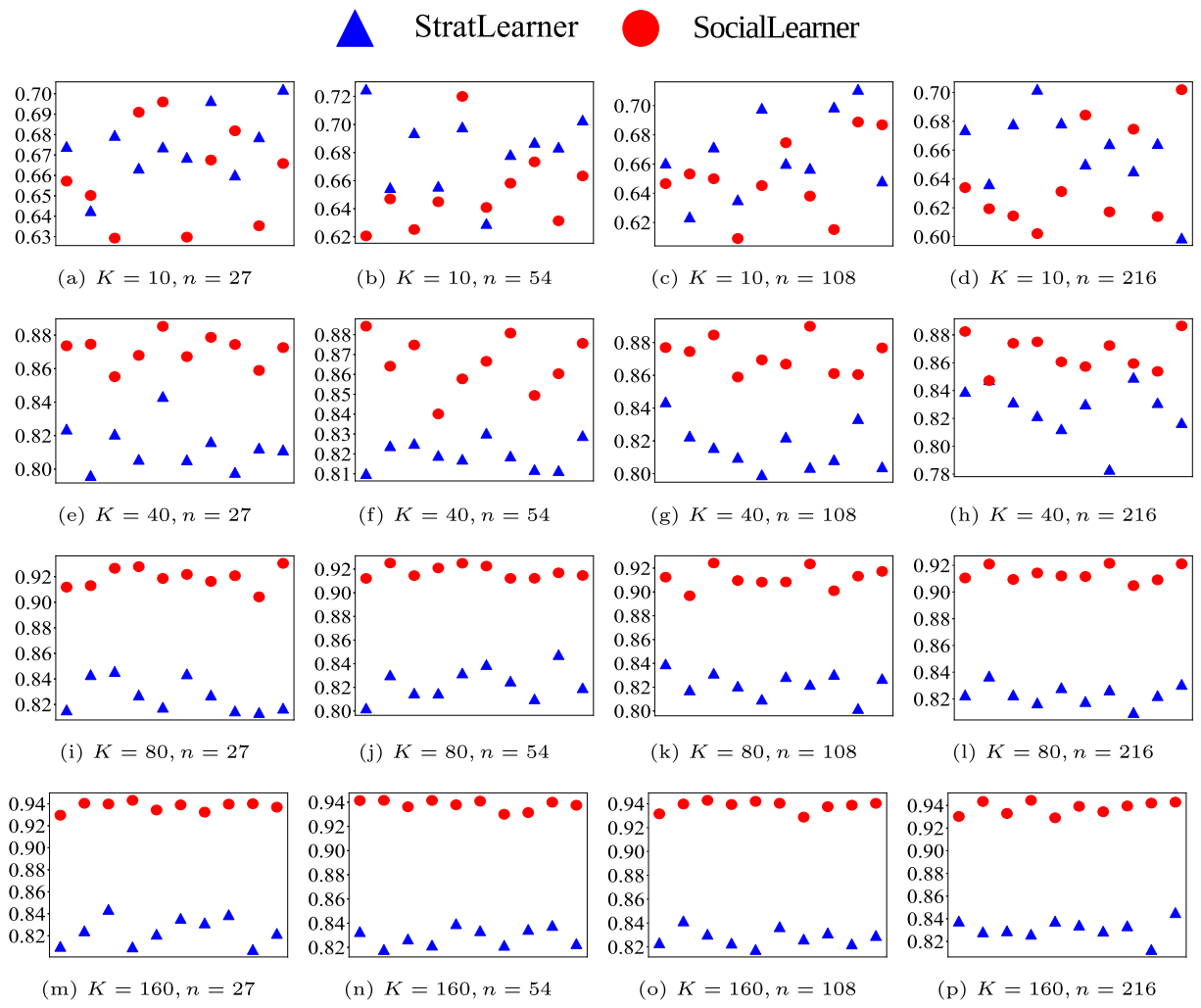


Fig. 1. The results on the ER graph without training. Each subgraph is labeled by the value of K and n and displays the results of 10 repeated experiments without training for StratLearner and SocialLearner. The result indicates the ratio $\frac{f(x,\hat{y},G)}{f(x,y,G)}$.

insufficient, the sampling method refined by SocialLearner does not necessarily perform better than the initial sampling method. However, with an increased number of sampled graphs, the quality of the sampling method refined by SocialLearner is improved significantly over the initial sampling method. Moreover, as the number of sampled graphs increases, the performance gap between SocialLearner and the initial method becomes larger. Another interesting observation is that increasing the number of sampled graphs from 100 to 400 significantly improves StratLearner's performance without training. However, further increasing the number of sampled graphs beyond this point results in less noticeable performance gains. This suggests that both StratLearner and SocialLearner require a sufficient number of samples to reach optimal performance levels.

Table 7
Results for Memory and Time Costs on ER Graph. The memory cost is measured in MB, and the time cost is measured in seconds. The total number of sampled graphs K' are 100, 400, 800, and 1600 respectively.

	Training Size n'	Memory Cost (MB) 100	400	800	1600	Time Cost (S) 100	400	800	1600
	270	51.1	195.5	387.2	772.0	111.0	435.31	987.3	2070.4
Strat-	540	53.1	197.9	390.1	776.4	665.2	197.9	1378.1	2677.3
Learner	1080	56.8	202.7	396.4	784.9	294.4	1154.5	2331.5	3848.5
	2160	64.4	212.2	408.5	802.2	518.5	2177.4	4356.2	6819.1
	270	8.9	26.5	52.8	144.9	94.1	364.9	691.6	1493.9
Social-	540	9.0	26.6	52.9	145.4	154.4	479.1	954.1	2069.4
Learner	1080	9.4	27.1	53.2	145.4	222.8	609.6	1437.6	3260.7
	2160	10.1	27.7	54.2	146.0	358.5	1153.1	2301.4	4764.6

Table 8
Results for the source detection on Twitter. Each cell shows the average ratio compared with the sample decision, together with the standard deviation (std). The total number of sampled graphs $K^{'}$ is 100, 400, 800, and 1600, respectively.

	Training	Number of sampled grpah			
	Size $n^{'}$	100	400	800	1600
USCO-solver	270	0.866 (2E−2)	0.895 (5E−2)	0.882 (4E−2)	0.908 (8E−2)
	540	0.833 (5E−2)	0.900 (7E−2)	0.881 (9E−2)	0.855 (9E−2)
	1080	0.886 (1E−1)	0.905 (8E−2)	0.887 (3E−2)	0.814(1E−2)
	2160	0.874 (7E−2)	0.870 (6E−2)	0.904 (6E−2)	0.866(9E−2)
Social-Learner	270	0.694 (2E−2)	0.870 (3E−2)	0.954 (4E−2)	0.979 (3E−2)
	540	0.677 (3E−2)	0.858 (2E−2)	0.980 (3E−2)	0.977 (2E−2)
	1080	0.687 (2E−2)	0.843 (1E−2)	0.950 (3E−2)	0.970 (2E−3)
	2160	0.714 (3E−2)	0.857 (3E−2)	0.939 (3E−2)	0.987 (2E−2)

We also report the time and memory cost of the StratLearner and SocialLearner in Table 7 for the misinformation prevention problem. It is not surprising that the memory cost of SocialLearner is less than that of StratLearner since SocialLearner trains models with less training data and fewer sampled subgraphs. However, we are pleased to find that the time cost of SocialLearner is also less than that of StratLearner, even though SocialLearner trains multiple small models while StratLearner trains a single model. These results demonstrate that SocialLearner is more efficient than StratLearner. Additionally, we observe that the number of sampled graphs has a greater impact on time cost than the amount of training data. Increasing the number of sampled graphs significantly improves performance but also increases time cost. Therefore, a trade-off between effectiveness and efficiency must be considered.

5.6. Real world application

To further validate the scalability and real-world applicability of our proposed framework, we conduct experiments on three investigated problems on the real-world Twitter dataset (De Domenico et al., 2013), which is recorded in the following format.

(post_id, post_time, reply_id), ...,(post_id, post_time, reply_id)),

where each tuple (post_id, post_time, reply_id) represents an interaction that the user reply_id forwarded a tweet originally posted by user post_id. We treat both post_id and reply_id as nodes in a graph. If a tuple exists between two nodes, we create a directed edge from the source to the forwarding user. The resulting graph consists of 1699 nodes and 2420 edges.

The diffusion time of an edge (u, v) is defined as the time difference between the earliest moment u received the tweet and the earliest moment v forwarded it from u. In addition, if an edge (u, v) appears multiple times, we select the smallest diffusion time. To formulate the propagation probability p, we first let $count(u, v)$ denote the number of times the edge (u, v) appears in the dataset. We then define p^* as the normalized frequency: $p^* = \frac{count(u,v)}{\max_{e \in E} count(e)}$. To reduce the sensitivity of p^* to high-frequency edges while preserving informative variation at lower frequencies, we apply a square root transformation: $p = \sqrt{p^*}$. The procedures for generating the dataset and constructing the sampled graph follow the same methodology described in Sec 5.2.

The results on the Twitter dataset are reported in Tables 8, 9, and 10. As expected, SocialLearner outperforms the baselines when provided with a sufficient number of sampled graphs. However, for source detection tasks, we observe that when the number of sampled graphs is small, SocialLearner performs worse than the USCO-solver. This highlights the importance of having enough sampled graphs to ensure robust performance, offering insight into practical deployment in real-world settings. Moreover, as the

Table 9
Results for the misinformation prevention on Twitter. Each cell shows the average ratio compared with the sample decision, together with the standard deviation (std). The total number of sampled graphs $K^{'}$ is 100, 400, 800, and 1600, respectively. OT indicates that training could not be completed within 4 h on our 128-core CPU server—more than twice the time required by SocialLearner.

	Training	Number of sampled grpah			
	Size $n^{'}$	100	400	800	1600
Strat-Learner	270	0.266 (5E−3)	0.325 (8E−3)	0.331 (9E−3)	OT
	540	0.265 (8E−3)	0.313 (1E−2)	0.315 (1E−2)	OT
	1080	0.267 (8E−3)	0.313 (2E−2)	0.323 (6E−3)	OT
	2160	0.271 (5E−3)	0.317 (5E−3)	0.322 (8E−3)	OT
Social-Learner	270	0.641 (5E−3)	0.912 (4E−3)	0.938 (9E−3)	0.947 (6E−3)
	540	0.650 (1E−2)	0.920 (1E−2)	0.943 (5E−3)	0.938 (9E−3)
	1080	0.642 (1E−2)	0.912 (1E−2)	0.930 (1E−2)	0.941 (1E−2)
	2160	0.634 (1E−2)	0.904 (1E−2)	0.934 (2E−2)	0.937 (1E−2)

Table 10
Results for the community protection on Twitter. Each cell shows the average ratio compared with the sample decision, together with the standard deviation (std). The total number of sampled graphs $K^{'}$ is 100, 400, 800, and 1600, respectively.

	Training	Number of sampled grpah			
	Size $n^{'}$	40	160	640	1600
USCO-solver	80	0.517 (2E−2)	0.565 (2E−2)	0.609 (1E−2)	0.598 (7E−3)
	160	0.535 (1E−2)	0.576 (2E−2)	0.607 (1E−2)	0.602 (2E−2)
	320	0.542 (1E−2)	0.554 (2E−2)	0.609 (1E−2)	0.639 (1E−2)
Social-Learner	80	0.586 (1E−2)	0.624 (1E−2)	0.634 (9E−3)	0.642 (1E−2)
	160	0.578 (1E−2)	0.628 (9E−3)	0.654 (1E−2)	0.647 (1E−2)
	320	0.654 (3E−2)	0.603 (8E−3)	0.674 (1E−2)	0.697 (1E−2)

graph size increases, the memory requirements and running time of StratLearner grow significantly, making it unable to find solutions within a reasonable time compared to SocialLearner. This further demonstrates the efficiency of SocialLearner.

6. Conclusion

In this paper, we introduce SocialLearner to solve the social influence risk management tasks. SocialLearner employs a two-step methodology inspired by ensemble learning, which enhances objective function estimation while reducing computational overhead without compromising accuracy. These advantages make SocialLearner a reliable alternative to current state-of-the-art approaches. Experiment results also demonstrate that SocialLearner outperforms current methods across multiple social networks and tasks. This highlights its potential for practical applications such as preventing misinformation, detecting sources, and protecting communities. For future research, we will explore generalizing SocialLearner to larger graphs, expanding its applicability to larger and more practical social structures, such as Facebook. Additionally, our current setting of generating training data selects nodes from a uniform distribution. An important direction for future research involves adapting SocialLearner to scenarios where nodes are sampled from imbalanced distributions, thereby better reflecting real-world social network characteristics.

CRediT authorship contribution statement

Siqi Wang: Writing – original draft, Methodology, Conceptualization. **Jiahao Xie:** Writing – review & editing. **Yifan Wang:** Writing – review & editing. **Guangmo Tong:** Supervision, Funding acquisition.

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.

Acknowledgement

This project is supported in part by the National Science Foundation under IIS-2144285 and IIS-2414308.

References

Adamic, L. A., Lukose, R. M., Puniyani, A. R., & Huberman, B. A. (2001). Search in power-law networks. *Physical Review E, 64*, Article 046135.
Azzimonti, M., & Fernandes, M. (2023). Social media networks, fake news, and polarization. *European Journal of Political economy, 76*, Article 102256.
Bakshy, E., Rosenn, I., Marlow, C., Adamic, L. (2012). The role of social networks in information diffusion, In: Proceedings of the 21st international conference on World Wide Web, 519-528.
Budak, C., Agrawal, D., El Abbadi, A. (2011). Limiting the spread of misinformation in social networks, In: Proceedings of the 20th international conference on World wide web, 665-674.
Cheng, L., Zhu, P., Tang, K., Gao, C., Wang, Z. (2024). Gin-sd: source detection in graphs with incomplete nodes via positional encoding and attentive fusion, In: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 55-63.
Demetrescu, C., Goldberg, A. V., Johnson, D.S. (2009). The shortest path problem: Ninth DIMACS implementation challenge, volume 74, American Mathematical Soc.
De Domenico, M., Lima, A., Mougel, P., & Musolesi, M. (2013). The anatomy of a scientific rumor. *Scientific Reports, 3*, 2980.
Dong, M., Zheng, B., Quoc Viet Hung, N., Su, H., Li, G. (2019). Multiple rumor source detection with graph convolutional networks, In: Proceedings of the 28th ACM international conference on information and knowledge management, 569-578.
Erdös, P., & Rényi, A. (1959). On random graphs i. *Publ math Debrecen, 6*, 18.
Fan, L., Lu, Z., Wu, W., Thuraisingham, B., Ma, H., Bi, Y. (2013). Least cost rumor blocking in social networks, In: 2013 IEEE 33rd International Conference on Distributed Computing Systems, IEEE, 540-549.
Farajtabar, M., Yang, J., Ye, X., Xu, H., Trivedi, R., Khalil, E., Li, S., Song, L., Zha, H. (2017). Fake news mitigation via point process based intervention, In: International conference on machine learning, PMLR, 1097-1106.
Freeman, L. (2004). The development of social network analysis. *A Study in the Sociology of Science, 1*, 159–167.
Guille, A., Hacid, H., Favre, C., & Zighed, D. A. (2013). Information diffusion in online social networks: a survey. *ACM Sigmod Record, 42*, 17–28.
He, X., Song, G., Chen, W., Jiang, Q. (2012). Influence blocking maximization in social networks under the competitive linear threshold model, In: Proceedings of the 2012 SIAM international conference on data mining, SIAM, 463-474.

Islam, M. R., Liu, S., Wang, X., & Xu, G. (2020). Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining, 10*, 82.
Joachims, T., Finley, T., & Yu, C.-N. J. (2009). Cutting-plane training of structural svms. *Machine Learning, 77*, 27–59.
Karger, D. R., & Stein, C. (1996). A new approach to the minimum cut problem. *Journal of the ACM (JACM), 43*, 601–640.
Kempe, D., Kleinberg, J., Tardos, É. (2003). Maximizing the spread of influence through a social network, In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 137-146.
Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., & Ghahramani, Z. (2010). Kronecker graphs: an approach to modeling networks. *Journal of Machine Learning Research, 11*.
Lewandowsky, S., & Van Der Linden, S. (2021). Countering misinformation and fake news through inoculation and prebunking. *European Review of Social Psychology, 32*, 348–384.
Li, M., Wang, X., Gao, K., & Zhang, S. (2017). A survey on information diffusion in online social networks: models and methods. *Information, 8*, 118.
Manchanda, S., Mittal, A., Dhawan, A., Medya, S., Ranu, S., & Singh, A. (2020). Gcomb: Learning budget-constrained combinatorial algorithms over billion-sized graphs. *Advances in Neural Information Processing Systems, 33*, 20000–20011.
Muhammed T, S., & Mathew, S. K. (2022). The disaster of misinformation: a review of research in social media. *International Journal of data Science and analytics, 13*, 271–285.
Nguyen, D. T., Nguyen, N. P., Thai, M. T. (2012). Sources of misinformation in online social networks: aaa to suspect?, In: MILCOM 2012-2012 IEEE Military Communications Conference, IEEE, , 1-6.
Nguyen, N. P., Yan, G., Thai, M. T., & Eidenbenz, S. (2012). Containment of misinformation spread in online social networks. *Proceedings of the 4th Annual ACM Web Science Conference,* 213–222.
Nguyen, N. P., Yan, G., & Thai, M. T. (2013). Analysis of misinformation containment in online social networks. *Computer Networks, 57*, 2133–2146.
Ognyanova, K., Lazer, D., Robertson, R. E., Wilson, C. (2020). Misinformation in action: Fake news exposure is linked to lower trust in media, higher trust in government when your side is in power, Harvard Kennedy School Misinformation Review.
Ping, Y., Cao, Z., Zhu, H. (2014). Sybil-aware least cost rumor blocking in social networks, In: 2014 IEEE Global Communications Conference, IEEE, 692-697.
Rice, E., Wilder, B., Onasch-Vera, L., Di Guiseppi, G., Petering, R., Hill, C., Yadav, A., Lee, S.-J., & Tambe, M. (2021). A peer-led, artificial intelligence–augmented social network intervention to prevent hiv among youth experiencing homelessness. *JAIDS Journal of Acquired Immune Deficiency Syndromes, 88*, S20–S26.
Shah, D., Zaman, T. (2010). Detecting sources of computer viruses in networks: theory and experiment, In: Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems, 203-214.
Shah, D., & Zaman, T. (2011). Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory, 57*, 5163–5181.
Shah, D., Zaman, T. (2012). Rumor centrality: a universal source detector, In: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems, 199-210.
Shelke, S., & Attar, V. (2019). Source detection of rumor in social network–a review. *Online Social Networks and Media, 9*, 30–42.
Stoer, M., & Wagner, F. (1994). *A simple min cut algorithm. European Symposium on Algorithms*. Springer141–147.
Su, K., Garg, P., Wu, W., & Du, D.-Z. (2023). Machine learning-based rumor controlling, handbook for management of threats: security and defense. *Resilience and Optimal Strategies,* 341–370.
Tabassum, S., Pereira, F. S., Fernandes, S., & Gama, J. (2018). Social network analysis: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8*, Article e1256.
Tong, G. A., Du, D.-Z. (2019). Beyond uniform reverse sampling: A hybrid sampling technique for misinformation prevention, In: IEEE INFOCOM 2019-IEEE conference on computer communications, IEEE, 1711-1719.
Tong, G. (2020). Stratlearner: Learning a strategy for misinformation prevention in social networks. *Advances in Neural Information Processing Systems, 33*, 15546–15555.
Tong, G. (2021). Usco-solver: Solving undetermined stochastic combinatorial optimization problems. *Advances in Neural Information Processing Systems, 34*, 1646–1659.
Tong, G. (2022). Social-inverse: Inverse decision-making of social contagion management with task migrations. *Advances in Neural Information Processing Systems, 35*, 31726–31740.
Torabi Asr, F., & Taboada, M. (2019). Big data and quality data for fake news and misinformation detection. *Big data Society, 6*, Article 2053951719843310.
Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y., & Singer, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research, 6*.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*.
Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. (2022). Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, In: International conference on machine learning, PMLR, 23965-23998.
Zadeh, N. (1972). Theoretical efficiency of the edmonds-karp algorithm for computing maximal flows. *Journal of the ACM (JACM), 19*, 184–192.
Zareie, A., & Sakellariou, R. (2021). Minimizing the spread of misinformation in online social networks: A survey. *Journal of Network and Computer Applications, 186*, Article 103094.
Zhang, H., Zhang, H., Li, X., & Thai, M. T. (2015). *Limiting the spread of misinformation while effectively raising awareness in social networks. Computational Social Networks: 4th International Conference, CSoNet 2015, Beijing, China, August 4-6, 2015, Proceedings 4*. Springer35–47.
Zhang, H., Alim, M. A., Li, X., Thai, M. T., & Nguyen, H. T. (2016). Misinformation in online social networks: Detect them all with a limited budget. *ACM Transactions on Information Systems (TOIS), 34*, 1–24.
Zhang, L., Ibrahim, S., & Fadzil, A. F. A. (2024). Rumor detection based on deep learning techniques: A systematic review. *TELKOMNIKA (Telecommunication Computing Electronics and Control), 22*, 869–886.
Zhu, K., & Ying, L. (2014). Information source detection in the sir model: A sample-path-based approach. *IEEE/ACM Transactions on Networking, 24*, 408–421.