Promoting Social Influence of Uncertain Seeds via Link Insertion

Xiaolong Chen
The Hong Kong University of Science and Technology
(Guangzhou)
Nansha, Guangzhou, China
xchen738@connect.hkust-gz.edu.cn

Jing Tang*
The Hong Kong University of Science and Technology
(Guangzhou)
The Hong Kong University of Science and Technology
jingtang@ust.hk

ABSTRACT

The emergence of link recommendation systems has triggered a line of research to study how to strategically insert links to enhance information diffusion in social networks. Existing literature assumes a seed set where all seed users are deterministically activated at the start of the campaign. However, uncertain seeding is increasingly prevalent and can be used to model more general scenarios like users' defaulting behavior or discount-based marketing. To investigate how to augment the influence of uncertain seeds by link recommendation, we formulate a problem named *influence maximization with augmentation for uncertain seeds* (IMAUS), which aims to insert k edges pointing from the uncertain seeds to others so as to maximize the influence of the given seeds. Due to the NP-hardness of the problem and the non-submodularity of the objective function, solving IMAUS is technically challenging. To address this issue, we resort to the sandwich strategy and propose two submodular bounding functions for the optimization objective. To overcome the #P-hardness of the bounding functions computation, we provide two unbiased estimators for the bounding functions via non-trivial usage of reverse influence sampling and devise greedy algorithms equipped with several accelerating techniques to return $(1-1/\mathrm{e}-\varepsilon)$-approximations for maximizing the bounding functions. With the above design, we instantiate the sandwich framework in a joint baking manner to reduce repeated sampling. Extensive experiments on 6 real-world datasets are conducted to validate the effectiveness and efficiency of the proposed methods. Specifically, our algorithm consistently produces higher influence increment than the baselines and is able to return a size-100 edge set for a billion-size graph within 10 minutes.

PVLDB Reference Format:
Xiaolong Chen and Jing Tang. Promoting Social Influence of Uncertain Seeds via Link Insertion. PVLDB, 19(1): XXX-XXX, 2026.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/chenxlong3/IMAUS.

1 INTRODUCTION

The concept of information diffusion is essential in social network analysis, where users are intricately connected, creating complex webs of relationships that enable the vast exchange of information, ideas, and opinions. Consequently, grasping and leveraging the underlying mechanisms of influence has become vital across various fields [15], including political campaigns and marketing [11]. An essential research issue in this domain is the concept of influence maximization (IM) [22], which involves determining the most effective subset of nodes in a social network to serve as the initiators of influence spread, consequently maximizing the expected number of influenced nodes. The seminal work by Kempe et al. [22] presents IM as a submodular optimization problem, sparking extensive research that addresses various aspects of the IM problem [3, 5–7, 13, 16, 19, 26, 31, 33, 34]. These studies primarily presuppose a static network topology.

Driven by the emergence of link recommendation methods such as Facebook's "People You May Know" and X's "Who to Follow," along with the extensive study of link recommendations for various objectives (e.g., minimizing polarization [1, 17, 43], maximizing pagerank fairness [36]), an increasing number of researchers investigate the impact of adding links to enhance influence spread [4, 8, 10, 20, 23, 38, 42]. However, most of the existing work assumes a set of deterministic seeds. That is, the seed nodes are initially active and will certainly propagate the influence.

A more realistic and general setting is to treat the activation of seeds as an uncertain event. For example, Yang et al. [40, 41] consider providing discounts for users. Each user has a discount-based probability to become activated and proceed to propagate the influence, which is related to the offered discount. On the other hand, Tong et al. [35] incorporate activation failure probabilities of the seeds, which is able to model users' potential default behavior. Link recommendation for such a general scenario is yet to be discussed despite its potential for a wide spectrum of applications.

Applications. Recommendation strategies risk inefficacy when seed users default on contractual obligations [35] by failing to post promotional content, and recommendation after users' activation may not be so profitable as users may exhibit limited engagement with the seed's prior posts. Modeling seeds as uncertain entities enables advertisers to design recommendation strategies that are robust against such defaults. On the other hand, e-commerce platforms (e.g., Amazon, Shopify) frequently use limited-time discounts [40, 41] to incentivize user participation. Integrating link recommendations with such discount strategies can amplify influence propagation by aligning structural network effects with economic incentives, thereby enhancing targeted marketing outcomes.

[*] Corresponding author: Jing Tang.

Proceedings of the VLDB Endowment, Vol. 19, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

Challenges. Promoting the influence of uncertain seeds via strategic link insertion presents dual challenges in problem formulation and methodology design. The problem takes into account three kinds of uncertainties: probabilistic seed activation, stochastic influence propagation, and probabilistic edge insertion. Such multi-faceted uncertainty complicates objective function analysis. From the perspective of methodology, since the objective function is not submodular, the performance of a naive greedy algorithm does not provide any non-trivial guarantee. In addition, for the link insertion scenario, the search space is larger than solely selecting nodes from the network. Overall, designing an efficient algorithm with theoretical guarantees is technically challenging.

Contributions. To promote the influence of uncertain seed users via link recommendation in a social network, we formalize the *influence maximization with augmentation for uncertain seeds* (IMAUS) problem, proving its NP-hardness and demonstrating that the objective function is neither submodular nor supermodular. To tackle the non-submodularity, we propose two submodular bounding functions, which allow us to leverage the sandwich framework [12, 37] to obtain a data-dependent approximation ratio for the problem. Despite the #P-hardness of bounding functions computation, we propose efficient unbiased estimators via a non-trivial application of reverse influence sampling, which allows us to maximize the bounding functions by integrating OPIM-C [31] with greedy algorithms on the estimation functions. However, naïvely running greedy algorithm is computationally expensive due to the large search space, and a direct application of OPIM-C is infeasible as it is designed for selecting influential nodes but not edges. To address these issues, we further design some acceleration techniques like marginal gain maintenance and low probability edge pruning to optimize the greedy edge selection. As for the sampling procedure, we carefully modify OPIM-C to fit our scenario. Moreover, we observe that our estimate functions can be jointly optimized by sharing the samples, thereby realizing the sandwich framework in a joint-baking manner [18]. Finally, we conduct extensive experiments on 6 real-world datasets to demonstrate the effectiveness and efficiency of the proposed algorithm.

To summarize, we make the following contributions.

1. We propose a new problem termed IMAUS and analyze its properties thoroughly. To our knowledge, we are the first to study IM-oriented link recommendation for uncertain seed users. (§3)
2. To provide an algorithm with a non-trivial approximation ratio through a sandwich framework, we establish two submodular bounds of the objective function. (§4)
3. We design efficient estimators on the proposed bounds and design several pruning strategies to efficiently maximize the proposed bounds, with which we instantiate the sandwich algorithm in an efficient joint baking manner. (§5)
4. Extensive experiments are conducted on 6 real-world graphs to demonstrate the effectiveness and efficiency of our approach. Specifically, our algorithm returns solutions with higher influence than other baselines on all tested datasets under various settings. As for efficiency, the proposed algorithm is able to finish within 10 minutes for Twitter, which contains over around 1.5 billion edges. (§6)

Table 1: Frequently used notations

Notation	Description
$G = (V, E)$	A social network with node set V and edge set E
n, m	The number of nodes and edges in G
E_C	The set of all candidate edges
k	The number of edges to be selected
S, A	The seed set, and the edge set
$\mathbf{p}$	The activation probability vector (aka. configuration)
$A^\circ, A^\circ_L, A^\circ_U$	The the optimal solution for $\sigma(\cdot, \mathbf{p})$, $\sigma^L(\cdot, \mathbf{p})$ and $\sigma^U(\cdot, \mathbf{p})$
$\sigma(A, \mathbf{p})$	The augmented influence spread of S after adding the edge set A
$\sigma^L(A, \mathbf{p}), \sigma^U(A, \mathbf{p})$	The lower bound and upper bound function
$\rho^L(A), \rho^U(A)$	$\sigma^L(A, \mathbf{p}) - \sigma(\mathbf{p})$, $\sigma^U(A, \mathbf{p}) - \sigma(\mathbf{p})$
$R, \mathcal{R}$	A random RR set, a collection of RR sets
$\mathcal{R}_v$	The collection of RR sets containing v
$\Gamma^L(A), \Gamma^U(A)$	The estimate of $\rho^L(A)$ and $\rho^U(A)$ (scaled by $|\mathcal{R}|/n$)

Due to the limited space, some proofs are omitted and could be found in the supplemental material of this paper.

2 PRELIMINARIES

In this section, we introduce necessary preliminaries for the studied problem, including graph terminologies and the concept of reverse influence sampling, a widely adopted technique in conventional IM. Frequently used notations are given in Table 1 for ease of reference.

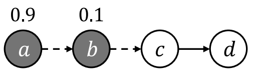
2.1 Influence Propagation

A social network is modeled as a directed graph $G = (V, E)$ with a node set V ($|V| = n$) and a directed edge set E ($|E| = m$). For an edge $\langle u, v \rangle \in E$, we call u an in-neighbor of v and v an out-neighbor of u. The in-neighbor (resp. out-neighbor) set of v is denoted by $N_{in}(v)$ (resp. $N_{out}(v)$). Each edge $\langle u, v \rangle$ is attached with an influence probability $p_{u,v} \in (0, 1]$, denoting the probability of u activating v.

Cascade model. In this paper, we adopt the widely-studied independent cascade (IC) model [3, 6, 22, 34]. The detail of influence propagation is illustrated as follows. At the initial timestamp 0, the chosen seed nodes are activated, leaving all other nodes inactive. When a node u first gets activated at timestamp i, it has a single opportunity to activate each node v of its inactive out-neighbors with probability $p_{u,v}$ at timestamp $i+1$, and it stays active for the duration of the propagation process. The diffusion process terminates when no additional nodes in the graph can be activated.

The information propagation process could be also described by *live edge* representation [22]. For the IC model, after removing each edge $\langle u, v \rangle \in E$ with probability $1 - p_{u,v}$, the remaining graph is referred to as a *realization* of G, which is denoted by ϕ. That is, for a particular $\phi \sim G$, $\Pr[\phi] = \prod_{e \in E_\phi} p_e \prod_{e \in E \setminus E_\phi} (1 - p_e)$. Let $I_\phi(S)$ be the number of nodes reachable from S in ϕ. The influence spread of a seed set S is defined as $\sum_{\phi \sim G} \Pr[\phi] I_\phi(S)$.

Influence of uncertain seeds. Probabilistic seeding is adopted to model consumer behaviors by previous literature [35, 40, 41]. Specifically, we assume a given probability vector $\mathbf{p} \in [0, 1]^n$ indicating the probability of each node being a deterministic seed and a node with non-zero entry in $\mathbf{p}$ is called an uncertain seed. Following prior work [40], we name $\mathbf{p}$ as a *configuration.* Mathematically, the probability that a subset $X \subseteq S$ is computed by



$$\sigma(\mathbf{p}) = 0.9 + 0.1 = 1$$
$$\sigma(\{e_1\}, \mathbf{p}) = 0.9 + 0.1 + 0.9 \cdot 0.9 = 1.81$$
$$\sigma(\{e_2\}, \mathbf{p}) = 0.9 + 0.1 + 2 \cdot 0.1 = 1.2$$
$$\sigma(\{e_1, e_2\}, \mathbf{p}) = 0.9 + (0.1 + 0.9 \cdot 0.9) \cdot 3 = 3.63$$

Figure 1: An example to illustrate the non-submodularity of IMAUS.

$\Pr[X] = \prod_{i \in X} \mathbf{p}_i \prod_{j \in S \setminus X}(1 - \mathbf{p}_j)$. Then considering ϕ as a realization of the uncertain graph G, the influence spread of a configuration $\mathbf{p}$ is defined as

$$\sigma(\mathbf{p}) = \sum_{\phi \sim G} \sum_{X \in 2^S} \Pr[\phi] \Pr[X] I_{\phi}(X), \tag{1}$$

As a generalized version of conventional influence function, the computation of $\sigma(\mathbf{p})$ is naturally #P-hard [5, 40].

Augmented influence via link insertion. Given an edge set $A \subseteq (S \times V) \setminus E$, the augmented influence spread of the configuration $\mathbf{p}$ after adding edge set A into the network is defined as

$$\sigma(A, \mathbf{p}) = \sum_{\phi \sim G} \sum_{X \in 2^S} \sum_{T \in 2^A} \Pr[\phi] \Pr[X] \Pr[T] I_{\phi(T)}(X), \tag{2}$$

where $\phi(T)$ is the resulting live edge graph after adding the edge set T into ϕ. For further convenience, we use A_u^+ to denote the edges in A that start from node u, and A_v^- to denote the edges in A that ends at node v. That is, $A_u^+ = \{\langle i, j \rangle \in A : i = u\}$ and $A_v^- = \{\langle i, j \rangle \in A : j = v\}$. For simplicity, we denote $\sigma(\emptyset, \mathbf{p})$ as $\sigma(\mathbf{p})$.

2.2 Reverse Influence Sampling

Borgs et al. [3] introduce a novel reverse influence sampling (RIS) framework for IM, which leverages sketch-based samples termed random *reverse reachable (RR) sets* to estimate seed set influence. A random RR set is generated through a two-step process:

(1) Select a node v from V uniformly at random.
(2) Sample a set R of the nodes in V, such that for any $u \in V$, the probability that it appears in R equals the probability that u can activate v in an influence propagation process.

With a slight abuse of notation, let $\sigma(S)$ denote the influence of the deterministic seed set S. Building on the notion of random RR sets, a key observation is established as follows.

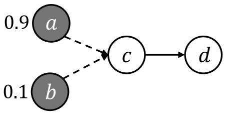
Lemma 2.1 ([3]). *For any seed set $S \in V$ and a random RR set R,*

$$\sigma(S) = n \cdot \Pr[S \cap R \neq \emptyset].$$

Given a set $\mathcal{R}$ of random RR sets, we say that a RR set $R \in \mathcal{R}$ is *covered* by a node set S if $R \cap S \neq \emptyset$. Denote by $\mathcal{R}_S$ the RR sets covered by S. Then, $\frac{n}{|\mathcal{R}|} |\mathcal{R}_S|$ is an unbiased estimate of $\sigma(S)$ according to Lemma 2.1. Based on RIS, Borgs et al. [3] propose a general framework for IM consisting of two steps. That is, we first (i) generate a set $\mathcal{R}$ of random RR sets, and then (ii) identify a node set S^* with the maximum coverage in $\mathcal{R}$ via a standard greedy algorithm. Utilizing RIS, the state-of-the-art IM algorithms [16, 19, 31, 33] have been proposed to reduce computation overheads.

3 PROBLEM FORMULATION

Conventional IMA problem aims to recommend deterministic seeds to ordinary nodes, while our problem extends it to the scenario of promoting the influence of uncertain seeds, whose formal definition is given by Definition 3.1.



$$\sigma(\mathbf{p}) = 0.9 + 0.1 = 1$$
$$\sigma(\{e_1\}, \mathbf{p}) = 0.9 + 0.1 + 2 \cdot 0.9 = 2.8$$
$$\sigma(\{e_2\}, \mathbf{p}) = 0.9 + 0.1 + 2 \cdot 0.1 = 1.2$$
$$\sigma(\{e_1, e_2\}, \mathbf{p}) = 0.9 + 0.1 + 2 \cdot (0.9 + 0.1 \cdot 0.1) = 2.82$$

Figure 2: An example to illustrate the non-supermodularity of IMAUS.

Definition 3.1 (Influence Maximization with Augmentation for Uncertain Seeds (IMAUS)). Given a graph $G = (V, E)$, a set of probabilistic seeds S, a configuration $\mathbf{p} \in \mathbb{R}^n$ and a candidate edge set $E_C \subseteq (S \times V) \setminus E$, IMAUS aims to select a size-k edge set $A \subseteq E_C$ to maximize the augmented influence spread $\sigma(A, \mathbf{p})$.

For the time complexity analysis throughout this paper, we consider the worst-case scale of the candidate edge set, i.e., $|E_C| \in \Theta(|S|\, n)$.

Computation Hardness. Since the proposed problem is a generalization of the NP-hard IMA problem, we have the following theorem to state the computation hardness of IMAUS.

Theorem 3.2. *IMAUS is NP-hard and the computation of $\sigma(A, \mathbf{p})$ is #P-hard.*

Moreover, we show that $\sigma(\cdot, \mathbf{p})$ is neither submodular nor supermodular under the IC model, making it hard to approximate.

Lemma 3.3. *$\sigma(A, \mathbf{p})$ is monotone but neither submodular nor supermodular w.r.t. A under the IC model.*

Proof. The monotonicity comes from the fact that adding edges does not decrease the number of reachable nodes from any given seed set. A counterexample for the non-submodularity is given in Figure 1. Suppose node a and b are two uncertain seed nodes with activation probability 0.9 and 0.1 respectively, and the influence probability on every edge is 1. $e_1 = \langle a, b \rangle$ and $e_2 = \langle b, c \rangle$ are two candidate edges to insert into the network. Obviously,

$$\sigma(\{e_1, e_2\}, \mathbf{p}) - \sigma(\{e_1\}, \mathbf{p}) = 1.82 > 0.2 = \sigma(\{e_2\}, \mathbf{p}) - \sigma(\mathbf{p}).$$

Therefore, $\sigma(\cdot, \mathbf{p})$ is not submodular. For the non-supermodularity, we have a counterexample shown in Figure 2. Similarly, we have

$$\sigma(\{e_1, e_2\}, \mathbf{p}) - \sigma(\{e_1\}, \mathbf{p}) = 0.02 < 0.2 = \sigma(\{e_2\}, \mathbf{p}) - \sigma(\mathbf{p}),$$

which violates the supermodularity. Hence the lemma holds. □

Also, we prove that the objective function is not even weakly-submodular and thus, a greedy algorithm fails to provide any nontrivial approximation ratio. The submodularity ratio is a metric to assess how closely the function approximates being submodular. Formally, for all $A, B \subset E_C$, the submodularity ratio is the largest scalar β with

$$\sum_{e \in A \setminus B} \left(\sigma(B \cup \{e\}, \mathbf{p}) - \sigma(B, \mathbf{p})\right) \geq \beta \left(\sigma(B \cup A, \mathbf{p}) - \sigma(B, \mathbf{p})\right). \tag{3}$$

Lemma 3.4. *$\sigma(A, \mathbf{p})$ is not weakly-submodular w.r.t. A.*

Proof. Consider a linked list with $V = \{v_0, v_1, \ldots, v_{n-1}\}$ and $E = \{\langle v_2, v_3 \rangle, \langle v_3, v_4 \rangle, \ldots, \langle v_{n-2}, v_{n-1} \rangle\}$. The influence probability of each edge (both existing and non-existing) is set to 1. The activation probability $\mathbf{p}_{v_0} = 1$ and $\mathbf{p}_{v_1}$ is a small enough number to be close to 0. Let $A = \{\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle\}$ and $B = \emptyset$. Then the left side

Algorithm 1 Sandwich Approximation Strategy

```
Input: A graph $G$, the budget $k$ and the parameters $\alpha_L, \alpha_U, \gamma$
Output: A size-$k$ edge set $A^*$
$A_U \leftarrow$ an $\alpha_U$-approximate solution to the upper bound function;
$A_L \leftarrow$ an $\alpha_L$-approximate solution to the lower bound function;
$A_H \leftarrow$ an arbitrary solution to the original function;
$\hat{\sigma}(\cdot, \mathbf{p}) \leftarrow$ a multiplicative $\gamma$-error estimate for $\sigma(\cdot, \mathbf{p})$
$A^* \leftarrow \arg\max_{A \in \{A_L, A_U, A_H\}} \hat{\sigma}(A, \mathbf{p})$
return $A^*$;
```

of Equation (3) is $(1 - \mathbf{p}_{v_1})\mathbf{p}_{v_0} + \mathbf{p}_{v_1}(n-2)$, and the right side is $\beta\left((1 - \mathbf{p}_{v_1})\mathbf{p}_{v_0}(n-1) + \mathbf{p}_{v_1}(n-2)\right)$. Since $\mathbf{p}_{v_1}$ is near 0, as n grows larger, the submodularity ratio β approaches 0 infinitely. Therefore, the objective function of IMAUS is not weakly-submodular. □

Discussion. The proposed IMAUS problem exhibits broad generalizability. On the one hand, when $\mathbf{p}$ satisfies that $\mathbf{p}_s = 1$ for all $s \in S$, IMAUS becomes IMA [8–10], which aims to insert links pointing from deterministic seeds to ordinary ones to augment the influence spread. On the other hand, the general link recommendation for IM where no constraint is imposed on the candidate edges can also be modeled by IMAUS since the source of each candidate edge has a probability to be activated. While the exact activation probability of a node is usually intractable, lower-bound approximations (e.g., maximum influence probability path [5] or hop-based computation [32]) enable practical initialization for the probabilities and transform the general link recommendation problem to IMAUS.

4 SANDWICH STRATEGY

4.1 Overview

Due to the non-submodularity of the objective function, it is hard to solve IMAUS with a constant approximation ratio. Fortunately, a data-dependent approximation ratio could be obtained by using a sandwich strategy, described in Algorithm 1. Specifically, with a submodular lower bound and a submodular upper bound for the objective function, the algorithm first find an α_L-approximate solution A_L and an α_U-approximate solution A_U to the lower bound and upper bound, respectively. Then it computes a heuristic solution A_H to the original problem. Finally, the best among A_L, A_U and A_H is returned as the final solution. A well-established result is that the solution A^* returned by the sandwich framework provides the following approximation guarantee.

$$\sigma(A^*, \mathbf{p}) \geq \max\left\{\frac{\sigma^L(A_L^\circ, \mathbf{p})}{\sigma(A^\circ, \mathbf{p})}\alpha_L, \frac{\sigma(A_U, \mathbf{p})}{\sigma^U(A_U, \mathbf{p})}\alpha_U\right\}\frac{1-\gamma}{1+\gamma}\sigma(A^\circ, \mathbf{p}), \quad (4)$$

where A_L° and A° are the optimal solutions for $\sigma^L(\cdot, \mathbf{p})$ and $\sigma(\cdot, \mathbf{p})$, respectively. Obviously, the sandwich framework relies on the bounds for the objective function. Next we present our design of monotone and submodular bounding functions for IMAUS.

4.2 Submodular Lower Bound

Intuitively, the non-submodularity comes from the fact that the activation probability of a candidate edge's source may be increased by the insertion of another edge. The rationale of designing the lower bound is that we only consider the influence increment of an edge $\langle s, v\rangle$ when the seed s is successfully initiated, ignoring the events where s is activated by other seeds. Following this, the lower bounding function is defined as

$$\sigma^L(A, \mathbf{p}) = \sum_{\phi \sim G}\sum_{X \in 2^S}\sum_{T \in 2^A} \Pr[\phi]\Pr[X]\Pr[T] I_{\phi(T_X)}(X), \quad (5)$$

where $T_X = \{\langle u, v\rangle \in T : u \in X\}$.

Lemma 4.1. *Given a configuration* $\mathbf{p}$ *and an edge set* A, $\sigma^L(A, \mathbf{p})$ *is monotone nondecreasing and submodular w.r.t.* A *under the IC model.*

Proof. For the monotonicity and submodularity, it suffices to prove that the function $\sum_{T \in 2^A} \Pr[T] I_{\phi(T_X)}(X)$ is monotone and submodular w.r.t. A. The monotonicity is trivial to prove as adding edges for fixed seed nodes does not decrease the number of reachable nodes.

$$\begin{aligned}\text{r.h.s. of Eq. (5)} &= \sum_{X \in 2^S}\Pr[X]\sum_{\phi \sim G}\sum_{T \in 2^A}\Pr[\phi]\Pr[T] I_{\phi(T_X)}(X) \\ &= \sum_{X \in 2^S}\Pr[X]\sum_{\phi \sim G}\sum_{T \in 2^{A_X^+}}\Pr[\phi]\Pr[T] I_{\phi(T)}(X) \\ &= \sum_{X \in 2^S}\Pr[X]\sigma(A_X, \mathbf{1}_X),\end{aligned}$$

where $A_X^+ = \{\langle u, v\rangle \in A : u \in X\}$, and $\mathbf{1}_X$ denotes a vector $\mathbf{p}$ with $\mathbf{p}_u = 1$ for $u \in X$ and $\mathbf{p}_u = 0$ for $u \notin X$. Given that $\sigma(A_X, \mathbf{1}_X)$ is submodular [10] w.r.t. A_X, it naturally holds that $\sigma^L(A, \mathbf{p})$ is also submodular. □

4.3 Submodular Upper Bound

Since the activation probability of any node is upper bounded by 1, in order to design the submodular upper bounding function, we may assume that the inserted edges are all from deterministic seeds, which means that when an inserted edge $\langle u, v\rangle$ is live, v is deemed activated. Formally, we construct the upper bound by

$$\sigma^U(A, \mathbf{p}) = \sum_{\phi \sim G}\sum_{X \in 2^S}\sum_{T \in 2^A} \Pr[\phi]\Pr[X]\Pr[T] I_\phi(X \cup S_T), \quad (6)$$

where $S_T = \{v : \langle u, v\rangle \in T\}$. The monotonicity and submodularity are supported by Lemma 4.2.

Lemma 4.2. *Given a configuration* $\mathbf{p}$ *and an edge set* A, $\sigma^U(A, \mathbf{p})$ *is monotone nondecreasing and submodular w.r.t.* A *under the IC model.*

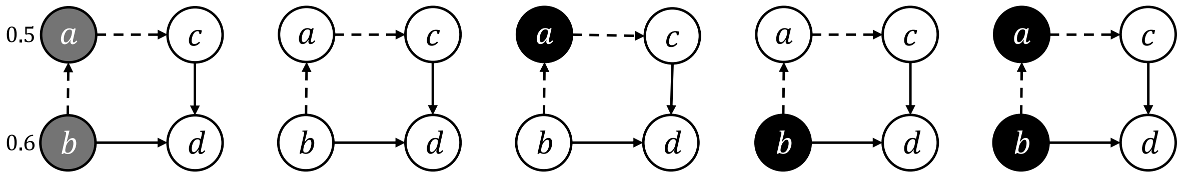
Proof. For the monotonicity, for any $B \supseteq A$,

$$\begin{aligned}\sigma^U(B, \mathbf{p}) = \sum_{\phi \sim G}\sum_{X \in 2^S}\sum_{T \in 2^A}\sum_{T' \in 2^{B \setminus A}} &\Pr[\phi]\Pr[X]\Pr[T]\Pr[T'] \\ &\cdot I_\phi(X \cup S_T \cup S_{T'}).\end{aligned}$$

As $\sum_{T' \in 2^{B \setminus A}} \Pr[T'] = 1$ and $I_\phi(X \cup S_T) \leq I_\phi(X \cup S_T \cup S_{T'})$, we have $\sigma^U(A, \mathbf{p}) \geq \sigma^U(B, \mathbf{p})$.

For the submodularity, since the non-negative linear combination of submodular functions is submodular, we consider fixed ϕ and X here. Then it suffices to prove that $\sum_{T \in 2^A} \Pr[T] I_\phi(X \cup S_T)$ is submodular w.r.t. A. Let $A \subseteq B$ and $e = \langle u, v\rangle$. For the ease of notation, let $I_\phi(v|X) = I_\phi(X \cup \{v\}) - I_\phi(X)$.

$$\begin{aligned}&\sum_{T \in 2^{A \cup \{e\}}}\Pr[T] I_\phi(X \cup S_T) - \sum_{T \in 2^A}\Pr[T] I_\phi(X \cup S_T) \\ &= \sum_{T \in 2^A}\Pr[T] p_{u,v} I_\phi(v|X \cup S_T).\end{aligned}$$



(a) Original Graph

(b) $X = \emptyset$ $\Pr[X] = 0.2$

(c) $X = \{a\}$ $\Pr[X] = 0.2$

(d) $X = \{b\}$ $\Pr[X] = 0.3$

(e) $X = \{a, b\}$ $\Pr[X] = 0.3$

Figure 3: An example to illustrate the bounding functions

For the edge set B,

$$\begin{aligned}&\sum_{T \in 2^{B \cup \{e\}}} \Pr[T] I_\phi(X \cup S_T) - \sum_{T \in 2^B} \Pr[T] I_\phi(X \cup S_T)\\ &= \sum_{T \in 2^B} \Pr[T] p_{u,v} I_\phi(v | X \cup S_T)\\ &= \sum_{T \in 2^A} \Pr[T] p_{u,v} \sum_{T' \in 2^{B \setminus A}} \Pr[T'] I_\phi(v | X \cup S_T \cup S_{T'}).\end{aligned}$$

Since $I_\phi(\cdot)$ is submodular, it naturally comes that $I_\phi(v | X \cup S_T \cup S_{T'}) \leq I_\phi(v | X \cup S_T)$. Hence, $\sigma^U(A, \mathbf{p})$ is submodular w.r.t. A. □

The hardness of computing the proposed bounding functions is given by Lemma 4.3. The proof is omitted as the result is inherited from the computation of $\sigma(\mathbf{p})$ [40, 41].

Lemma 4.3. *Both computing $\sigma^U(\cdot, \mathbf{p})$ and $\sigma^L(\cdot, \mathbf{p})$ are #P-hard.*

Example 4.4. Consider the graph given in Figure 3a, consisting of four nodes (a, b, c and d) and two edges ($\langle b, d\rangle$ and $\langle c, d\rangle$). The seed set is set as $\{a, b\}$, with activation probability $\mathbf{p}_a = 0.5$ and $\mathbf{p}_b = 0.6$. Suppose an edge set $A = \{\langle b, a\rangle, \langle a, c\rangle\}$ is to be inserted into the network and the influence probability of each edge is fixed as 1. The original objective function $\sigma(A, \mathbf{p})$ can be simply computed by $0.2 \cdot 0 + 0.2 \cdot 3 + 0.3 \cdot 4 + 0.3 \cdot 4 = 3$. For the lower bound function, each inserted edge is only taken into account when the source of it is successfully activated as a seed. Therefore, in Figure 3c, $\langle a, c\rangle$ will not be inserted into the graph. Then the lower bounding function $\sigma^L(A, \mathbf{p}) = 0.2 \cdot 0 + 0.2 \cdot 3 + 0.3 \cdot 3 + 0.3 \cdot 4 = 2.7$. For the upper bound function, recall that for every live newly inserted edge, we assume the target node of it will be activated in that realization. As a result, for the case of Figure 3b, nodes a and c will be considered activated by the two inserted edges. Hence, the upper bound $\sigma^U(A, \mathbf{p}) = 0.2 \cdot 3 + 0.2 \cdot 3 + 0.3 \cdot 4 + 0.3 \cdot 4 = 3.6$.

The effectiveness of the sandwich framework essentially relies on tight submodular bounding functions. To provide an intuition on the tightness of the proposed bounds, we discuss equality conditions between them and the objective function: When the seeds are deterministically activated (i.e., $\mathbf{p} = \mathbf{1}_S$), we have $\sigma^L(A, \mathbf{1}_S) = \sigma(A, \mathbf{1}_S) = \sigma^U(A, \mathbf{1}_S)$. When the uncertain seeds cannot be activated by other nodes (i.e., every uncertain seed has no incoming edge or candidate edge), $\sigma^L(A, \mathbf{p}) = \sigma(A, \mathbf{p})$.

5 EFFICIENT IMPLEMENTATION

With the newly proposed submodular bounding functions, the key challenge lies in efficiently instantiating the sandwich framework. In this section, we begin by introducing the overall joint baking framework that uses the same set of samples to optimize both bounding functions (§5.1), and then detail the maximization of the bounding functions, consisting of efficient estimators (§5.2), refined greedy selection algorithms (§5.3) and a modified sampling schedule (§5.4). Finally, we put it together and provide theoretical analysis on the proposed algorithm (§5.5).

Algorithm 2 JB-PIUS

```
Input: A graph $G$, the budget $k$ and the parameters $\alpha_1, \alpha_2, \gamma$
Output: A size-$k$ edge set $A^*$
$\theta_0 \leftarrow$ initial number of samples;
$\theta_{\max} \leftarrow$ the worst-case number of samples to return guaranteed solutions;
$i_{\max} \leftarrow \lceil \log_2 \frac{\theta_{\max}}{\theta_0} \rceil$
generate two collections of random RR sets $\mathcal{R}_1$ and $\mathcal{R}_2$ with $|\mathcal{R}_1| = |\mathcal{R}_2| = \theta_0$;
$\alpha_L \leftarrow 0$; $\alpha_U \leftarrow 0$;
for $i \leftarrow 1$ to $i_{\max}$ do
    if $\alpha_L < 1 - 1/e - \varepsilon$ then
        $A_L \leftarrow$ select the edge set by Algorithm 3
        $\alpha_L \leftarrow$ compute the empirical guarantee for $\rho^L(\cdot)$;
    if $\alpha_U < 1 - 1/e - \varepsilon$ then
        $A_U \leftarrow$ select the edge set by Algorithm 4
        $\alpha_U \leftarrow$ compute the empirical guarantee for $\rho^U(\cdot)$;
    if both $\alpha_L$ and $\alpha_U$ are larger than $1 - 1/e - \varepsilon$ then
        break;
    double the size of $\mathcal{R}$ by generating new samples;
$A_H \leftarrow$ an arbitrary heuristic solution to the original function;
$\hat{\sigma}(\cdot, \mathbf{p}) \leftarrow$ a multiplicative $\gamma$-error estimate for $\sigma(\cdot, \mathbf{p})$
$A^* \leftarrow \arg\max_{A \in \{A_L, A_U, A_H\}} \hat{\sigma}(A, \mathbf{p})$
return $A^*$;
```

5.1 A Joint Baking Framework

For the purpose of efficient implementation, we instantiate the aforementioned sandwich framework in a joint baking manner [18], where the bounding functions are maximized using the same set of samples, which brings much efficiency by avoiding repeated sampling. The proposed algorithm is called JB-PIUS (**J**oint **B**aking algorithm to **P**romote the **I**nfluence of **U**ncertain **S**eeds), the pseudocode of which is given in Algorithm 2. Based on the idea of OPIM-C [31], two collections of RR sets will be sampled to compute the empirical approximation ratio of the bounding functions. JB-PIUS first determines the initial and worse-case sample size. In each iteration of the main loop, we maximize the bounding functions by running well-designed greedy algorithms on the sampled RR sets, and test whether the empirical guarantees (α_L and α_U) exceed $1 - 1/e - \varepsilon$. We exit the loop if both α_L and α_U are larger than $1 - 1/e - \varepsilon$. Otherwise, the sample size will be doubled with newly sampled RR sets. After the loop, we have two approximate solutions A_L and A_U, for the lower and upper bound function, respectively. Use A_H to denote any heuristic solution to the original problem. We will finally compare the quality of A_H, A_L and A_U with a multiplicative γ-error estimator and select the best one among them. Next we will delve into the details of obtaining approximate solutions for both bounding functions.

5.2 Bounding Functions Estimation

As the approximation ratio for submodular optimization mainly focuses on normalized functions, i.e., $f(\emptyset) = 0$, we will discuss the normalized bounding functions in the following, denoted as $\rho^U(A, \mathbf{p}) = \sigma^U(A, \mathbf{p}) - \sigma(\mathbf{p})$ and $\rho^L(A, \mathbf{p}) = \sigma^L(A, \mathbf{p}) - \sigma(\mathbf{p})$.

To efficiently maximize the bounding functions that are #P-hard to compute, we should first investigate how to estimate the bounding functions by using the sampled RR sets. For both the upper and lower bound estimation, the intuition is to compute the probability that a random RR set changes its state from *not covered* to *covered* by the given seed set. To estimate the normalized influence increment of a given edge set, we should first eliminate the effect of the existence of uncertain seeds. Under the setting of conventional IM, each RR set can be abstracted as a Bernoulli random variable equal to 0 or 1, and the initial utility value of each RR set is 1 as no RR set has been covered yet. However, in our case of uncertain seeds, each RR set is initially covered with a certain probability by the seeds. So in order to estimate the normalized functions, we should initialize the utility value of each RR set by computing the probability that an RR set is not covered by any seed node. Specifically, the initial utility value of an RR set R is defined as

$$\pi_0(R) = \prod\nolimits_{u \in R}(1 - \mathbf{p}_u), \tag{7}$$

which could be computed and stored during the sampling phase. Then we delve into the formulation of the estimators for the proposed bounding functions.

Lower bound estimation. Now consider the lower bound function, which only counts the contribution of the edges pointing from successfully activated seeds. That is, for any edge $\langle u, v\rangle \in A$, we only consider the insertion of it when u is successfully activated as a seed. Intuitively, for a particular RR set R, the event that R changes its state from uncovered to covered could be considered as the intersection of the following two events - (i) R is not covered by any successfully activated seed, and (ii) there is at least one successfully activated seed u, such that at least one edge in A_u^+ activates R. Following this intuition, we derive an unbiased estimator for the lower bound function, formalized by Lemma 5.1.

Lemma 5.1. *Let R be a random RR set, we have*

$$\rho^L(A) = n \cdot \mathbb{E}\left[(1 - \prod\nolimits_{u \in S\backslash R}(1 - \mathbf{p}_u w(u, R, A)))\pi_0(R)\right], \tag{8}$$

where $w(u, R, A) = 1 - \prod_{\langle u,v\rangle \in A_u^+}(1 - p_{u,v}\mathbb{I}(v \in R))$.

Proof. Let $S_{T,X}$ be $\{v \in V : \langle u, v\rangle \in T \text{ and } u \in X\}$. Then by the rationale of RIS, we have

$$\rho^L(A) = \sigma^L(A, \mathbf{p}) - \sigma(\mathbf{p})$$

$$= \sum_{X \in 2^S, T \in 2^A} \Pr[X] \Pr[T] \sum_{v \in V} \Pr[(X \nrightarrow v) \wedge (S_{T,X} \rightarrow v)]$$
$$= n \cdot \mathbb{E}_R \mathbb{E}_X \mathbb{E}_T[\mathbb{I}(R \cap X = \emptyset)\mathbb{I}(R \cap S_{T,X} \neq \emptyset)].$$

Since $R \cap S_{T,X} \neq \emptyset$ indicates that at there is least one seed $u \in X$, such that at least one $\langle u, v\rangle \in A_u^+$ satisfies that $\langle u, v\rangle \in T$ and $v \in R$, we have

$$\begin{aligned}
&\mathbb{I}(R \cap S_{T,X} \neq \emptyset)\\
=&1 - \prod_{u \in X} \prod_{\langle u,v\rangle \in A_u^+} (1 - \mathbb{I}(\langle u, v\rangle \in T)\mathbb{I}(v \in R))\\
=&1 - \prod_{u \in S}(1 - \mathbb{I}(u \in X)(1 - \prod_{\langle u,v\rangle \in A} (1 - \mathbb{I}(\langle u, v\rangle \in T)\mathbb{I}(v \in R)))).
\end{aligned}$$

The correctness of the second equality could be easily verified by considering two events, $u \in X$ and $u \notin X$. Define $W_T(u, R, A) = (1 - \prod_{\langle u,v\rangle \in A_u^+}(1 - \mathbb{I}(\langle u, v\rangle \in T)\mathbb{I}(v \in R)))$ for convenience and $\mathbb{E}_T[W_T(u, R, A)] = w(u, R, A)$. We have

$$\begin{aligned}
&\mathbb{I}(R \cap X = \emptyset)\mathbb{I}(R \cap S_{T,X} \neq \emptyset)\\
=&\prod_{u \in R} \mathbb{I}(u \notin X) \cdot (1 - \prod_{u \in S}(1 - \mathbb{I}(u \in X)W_T(u, R, A)))\\
=&\prod_{u \in R} \mathbb{I}(u \notin X) \cdot (1 - \prod_{u \in S\backslash R} (1 - \mathbb{I}(u \in X)W_T(u, R, A))).
\end{aligned}$$

The first equality is obtained by the definition of $R \cap X \neq \emptyset$, and the second equality holds since if $\mathbb{I}(R \cap X = \emptyset)$, for every $u \in R$, u should not be in X, and if $\mathbb{I}(R \cap X \neq \emptyset)$, the result is 0, same for both side. Combining the linearity of expectation with the following two facts - (i) seed activations are independent of edge activations and (ii) the activation of each seed/edge is independent from one another, we have

$$\begin{aligned}
&\mathbb{E}_R\mathbb{E}_X\mathbb{E}_T\left[\prod_{u \in R} \mathbb{I}(u \notin X) \cdot (1 - \prod_{u \in S\backslash R} (1 - \mathbb{I}(u \in X)W_T(u, R, A)))\right]\\
=&\mathbb{E}_R\mathbb{E}_X\left[\prod_{u \in R} \mathbb{I}(u \notin X) \cdot (1 - \prod_{u \in S\backslash R} (1 - \mathbb{I}(u \in X)\mathbb{E}_T[W_T(u, R, A)]))\right]\\
=&\mathbb{E}_R\left[\prod_{u \in R} \mathbb{E}_X[\mathbb{I}(u \notin X)] \cdot (1 - \prod_{u \in S\backslash R} (1 - \mathbb{E}_X[\mathbb{I}(u \in X)]w(u, R, A)))\right]\\
=&\mathbb{E}_R[\pi_0(R) \cdot (1 - \prod\nolimits_{u \in S\backslash R}(1 - \mathbf{p}_u w(u, R, A)))].
\end{aligned}$$

Hence the lemma holds. □

Define

$$\Gamma^L(A) = \sum\nolimits_{R \in \mathcal{R}}(1 - \prod\nolimits_{u \in S\backslash R}(1 - \mathbf{p}_u w(u, R, A)))\pi_0(R). \tag{9}$$

Then with a collection $\mathcal{R}$ of random RR sets, $\frac{n\Gamma^L(A)}{|\mathcal{R}|}$ serves as an unbiased estimate for $\rho^L(A)$.

Upper bound estimation. For the upper bound function, all inserted edges are assumed to be pointed from deterministic seeds. Hence, for any inserted edge $\langle u, v\rangle$, RR sets containing v have an additional probability of $p_{u,v}$ to be covered. Then considering all edges in A, the probability of an RR set changes its state (from

Algorithm 3 Edge Selection for Lower Bound Maximization

```
Input: $E_C, \mathcal{R}, \mathbf{p}, k$
Output: A size-$k$ edge set $A^*$
$A^* \leftarrow \emptyset$; $S_v \leftarrow \emptyset$ for $v \in V$;
for $\langle u, v \rangle \in E_C$ do
    Add $u$ to $S_v$;
    $\Delta(\langle u, v \rangle) \leftarrow \mathbf{p}_u p_{u,v} \cdot \sum_{R \in \mathcal{R}_v \setminus \mathcal{R}_u} \pi_0(R)$;
initialize $\pi(R) \leftarrow \pi_0(R)$ for all $R \in \mathcal{R}$;
for $i = 1$ to $k$ do
    $\langle u^*, v^* \rangle \leftarrow \arg\max_{\langle u,v \rangle \in E_C \setminus A^*} \Delta(\langle u, v \rangle)$;
    $A^* \leftarrow A^* \cup \{\langle u^*, v^* \rangle\}$;
    for $R \in \mathcal{R}_{v^*} \setminus \mathcal{R}_{u^*}$ do
        for $v \in R$ do
            for $u \in S_v \setminus R$ do
                if $u = u^*$ then
                    $a \leftarrow \mathbf{p}_u p_{u,v} \frac{(1-w(u,R))}{(1-\mathbf{p}_u w(u,R))}$;
                    $b \leftarrow p_{u^*,v^*}$;
                    $\Delta(\langle u, v \rangle) \leftarrow \Delta(\langle u, v \rangle) - \pi(R) \cdot a \cdot b$;
                else
                    $a \leftarrow \mathbf{p}_u p_{u,v} \frac{(1-w(u,R))}{(1-\mathbf{p}_u w(u,R))}$;
                    $b \leftarrow \mathbf{p}_{u^*} p_{u^*,v^*} \frac{(1-w(u^*,R))}{(1-\mathbf{p}_{u^*} w(u^*,R))}$
                    $\Delta(\langle u, v \rangle) \leftarrow \Delta(\langle u, v \rangle) - \pi(R) \cdot a \cdot b$;
        $\pi(R) \leftarrow \pi(R) / (1 - \mathbf{p}_{u^*} w(u^*, R))$;
        $w(u^*, R) \leftarrow w(u^*, R) + (1 - w(u^*, R)) p_{u^*,v^*}$;
        $\pi(R) \leftarrow \pi(R)(1 - \mathbf{p}_{u^*} w(u^*, R))$;
return $A^*$;
```

not covered to covered) is $(1 - \prod_{v \in R} \prod_{\langle u,v \rangle \in A_v^-} (1 - p_{u,v}))$. Taking $\pi_0(R)$ into account, the estimator of the upper bound is then formalized by Lemma 5.2.

Lemma 5.2. *Let R be a random RR set. For any edge set A, we have*

$$\rho^U(A) = n \cdot \mathbb{E}\left[(1 - \prod_{v \in R} \prod_{\langle u,v \rangle \in A_v^-} (1 - p_{u,v}))\pi_0(R)\right]. \quad (10)$$

Proof. Similar to the proof of lower bound estimation, we have

$$\begin{aligned} \sigma^U(A, \mathbf{p}) - \sigma(\mathbf{p}) &= \mathbb{E}_R \mathbb{E}_X \mathbb{E}_T [\mathbb{I}(R \cap X = \emptyset) \mathbb{I}(R \cap S_T \neq \emptyset)] \\ &= \mathbb{E}_R \left[\mathbb{E}_X [\mathbb{I}(R \cap X = \emptyset)] \mathbb{E}_T [\mathbb{I}(R \cap S_T \neq \emptyset)]\right] \\ &= \mathbb{E}_R \left[\mathbb{E}_X \left[\prod_{u \in R} \mathbb{I}(u \notin X)\right] \mathbb{E}_T [\mathbb{I}(R \cap S_T \neq \emptyset)]\right] \\ &= \mathbb{E}_R [\pi_0(R) \mathbb{E}_T [\mathbb{I}(R \cap S_T \neq \emptyset)]]. \end{aligned}$$

The second equality is due to the fact that the activation of uncertain seeds is independent of the existence of inserted edges, and the third equality comes from the definition of $R \cap X = \emptyset$. The fourth equality is obtained by the linearity of expectation and the independence of each seed's activation. Now we analyze the computation of $\mathbb{E}_T [\mathbb{I}(R \cap S_T \neq \emptyset)]$.

$$\begin{aligned} \mathbb{E}_T [\mathbb{I}(R \cap S_T \neq \emptyset)] &= \mathbb{E}_T [1 - \prod_{v \in R} \mathbb{I}(v \notin S_T)] \\ &= \mathbb{E}_T [1 - \prod_{v \in R} \prod_{\langle u,v \rangle \in A_v^-} (1 - \mathbb{I}(\langle u, v \rangle \in T))] \\ &= 1 - \prod_{v \in R} \prod_{\langle u,v \rangle \in A_v^-} (1 - p_{u,v}) \end{aligned}$$

Putting together completes the proof. □

Algorithm 4 Edge Selection for Upper Bound Maximization

```
Input: $E_C, \mathcal{R}, \mathbf{p}, k$
Output: A size-$k$ edge set $A^*$
$A^* \leftarrow \emptyset$; $S_v \leftarrow \emptyset$ for $v \in V$;
for $\langle u, v \rangle \in E_C$ do
    Add $u$ to $S_v$;
for $v \in V \setminus S$ do
    sort $S_v$ by descending order of $p_{u,v}$;
    $u \leftarrow S_v$.top(); $\Delta(v) \leftarrow p_{u,v} \cdot \sum_{R \in \mathcal{R}_v} \pi_0(R)$;
initialize $\pi(R) \leftarrow \pi_0(R)$ for all $R \in \mathcal{R}$;
for $i = 1$ to $k$ do
    $v^* \leftarrow \arg\max_v \Delta(v)$; $u^* \leftarrow S_{v^*}$.pop();
    $A^* \leftarrow A^* \cup \{\langle u^*, v^* \rangle\}$;
    for $R \in \mathcal{R}_{v^*}$ do
        for $v \in R$ do
            $u \leftarrow S_v$.top(); $\Delta(v) \leftarrow \Delta(v) - \pi(R) \cdot p_{u^*,v^*} \cdot p_{u,v}$;
        $\pi(R) \leftarrow \pi(R)(1 - p_{u^*,v^*})$;
    $u \leftarrow S_{v^*}$.top(); $\Delta(v^*) \leftarrow \frac{p_{u,v^*}}{p_{u^*,v^*}} \cdot (1 - p_{u^*,v^*}) \cdot \Delta(v^*)$;
return $A^*$;
```

Defining

$$\Gamma^U(A) = \sum_{R \in \mathcal{R}} (1 - \prod_{v \in R} \prod_{\langle u,v \rangle \in A_v^-} (1 - p_{u,v})) \pi_0(R), \quad (11)$$

$\frac{n\Gamma^U(A)}{|\mathcal{R}|}$ brings an unbiased estimate for $\rho^U(A)$ after sampling a collection of random RR sets $\mathcal{R}$.

It is worth noting that both the lower and upper bound can be estimated within the same sample space, which builds the foundation of the joint baking framework to eliminate redundant sampling. In the following, details of optimization on the bounding functions will be introduced.

5.3 Greedy Selection

With the unbiased estimator above, a $(1 - 1/e)$ approximation on $\Gamma^L(\cdot)$ (resp. $\Gamma^U(\cdot)$) yields a $(1 - 1/e - \varepsilon)$-approximation for the maximization of $\rho^L(\cdot)$ (resp. $\rho^U(\cdot)$) given enough samples [33]. Now we show that such a guarantee could be obtained via a simple greedy algorithm by proving that both $\Gamma^L(\cdot)$ and $\Gamma^U(\cdot)$ are monotone non-decreasing and submodular.

Lemma 5.3. *$\Gamma^L(A)$ and $\Gamma^U(A)$ are monotone non-decreasing and submodular w.r.t. A.*

However, if we compute $\Gamma^L(\cdot)$(or $\Gamma^U(\cdot)$) naively in every iteration, the greedy algorithm takes up to $O(k\,|S|\,n \sum_{R \in \mathcal{R}} |R|)$ time, which is prohibitive. So we incorporate the following techniques to accelerate the implementation.

Lazy update. We first employ a widely-adopted acceleration technique in submodular maximization, namely CELF [26]. Specifically, for a submodular function $\Gamma(A)$, if $\Gamma(e_1|A) \geq \Gamma(e_2|A')$ for $A' \subseteq A$, $\Gamma(e_1|A) \geq \Gamma(e_2|A)$ naturally holds, where $\Gamma(e|A) = \Gamma(A \cup \{e\}) - \Gamma(A)$. Hence, if the marginal gain of an edge remains the largest after updating, there is no need to evaluate the marginal gain of the rest candidate edges.

Marginal gain maintenance. To reduce redundant computation, a natural idea is to maintain the marginal gain of each candidate element. However, such maintenance requires rigorous analysis. Let $\langle u^*, v^*\rangle$ be the edge to be inserted into A and $\langle u', v'\rangle$ be any other edge in $E_C \setminus A$. We have the following lemma for $\Gamma^L(\cdot)$.

Lemma 5.4. *Use $\Gamma^L(e|A)$ to denote $\Gamma^L(A \cup \{e\}) - \Gamma^L(A)$. We have*

$$\begin{aligned}&\Gamma^L(\langle u', v'\rangle \mid A \cup \{\langle u^*, v^*\rangle\}) \\ =&\Gamma^L(\langle u', v'\rangle \mid A) - \sum_{R \in \mathcal{R}_{v^*} \cap \mathcal{R}_{v'} \setminus \mathcal{R}_{u'} \setminus \mathcal{R}_{u^*}} \pi(R) \cdot a \cdot b,\end{aligned}$$

where $\pi(R) = \pi_0(R) \prod_{u \in S \setminus R}(1 - \mathbf{p}_u w(u, R, A))$, $a = \mathbf{p}_{u'} p_{u',v'} \cdot \frac{1 - w(u', R, A)}{1 - \mathbf{p}_{u'} w(u', R, A)}$ *and*

$$b = \begin{cases} p_{u^*,v^*} & \text{, if } u' = u^* \\ \mathbf{p}_{u^*} p_{u^*,v^*} \frac{1 - w(u^*, R, A)}{1 - \mathbf{p}_{u^*} w(u^*, R, A)} & \text{, else.} \end{cases}$$

Then for the greedy algorithm on $\Gamma^L(\cdot)$, we first initialize the marginal gain of each edge as $\Delta(\langle u, v\rangle) = \mathbf{p}_u p_{u,v} \sum_{R \in \mathcal{R}_v \setminus \mathcal{R}_u} \pi_0(R)$, where $\mathcal{R}_v = \{R \in \mathcal{R} : v \in R\}$. In each iteration, after selecting the edge with the highest marginal gain, denoted by $\langle u^*, v^*\rangle$, we traverse the RR sets in $\mathcal{R}_v \setminus \mathcal{R}_u$. For each node v in the RR set, we update the marginal gain of each $\langle u, v\rangle \in E_C$ that ends at v with $u \notin R$ by subtracting $\Delta(\langle u, v\rangle)$ by $\pi(R) \cdot a \cdot b$, with a and b defined by Lemma 5.4. After that, we need to update the value of $\pi(R)$ by

$$\pi(R) \leftarrow \pi(R) \cdot \frac{1 - \mathbf{p}_{u^*} w(u^*, R, A \cup \{\langle u^*, v^*\rangle\})}{1 - \mathbf{p}_{u^*} w(u^*, R, A)}.$$

Next we turn to the marginal gain maintenance for $\Gamma^U(\cdot)$, which is characterized by Lemma 5.5.

Lemma 5.5. *Use $\Gamma^U(e|A)$ to denote $\Gamma^U(A \cup \{e\}) - \Gamma^U(A)$. Then*

$$\begin{aligned}&\Gamma^U(\langle u', v'\rangle \mid A \cup \{\langle u^*, v^*\rangle\}) \\ =&\Gamma^U(\langle u', v'\rangle \mid A) - \sum_{R \in \mathcal{R}_{v^*} \cap \mathcal{R}_{v'}} p_{u^*,v^*} p_{u',v'} \cdot \pi(R),\end{aligned}$$

where $\pi(R) = \pi_0(R) \cdot \prod_{v \in R} \prod_{\langle u,v\rangle \in A_v^-} (1 - p_{u,v})$.

With Lemma 5.5, we detail the greedy selection algorithm for the upper bound as follows. The marginal gain of every edge $\langle u, v\rangle \in V$ is initialized as $p_{u,v} \cdot \sum_{R \in \mathcal{R}_v} \pi_0(R)$. After selecting the edge with the highest marginal gain $\langle u^*, v^*\rangle$, we traverse the RR sets containing v^* (denoted by $\mathcal{R}_{v^*}$). Given an RR set $R \in \mathcal{R}_{v^*}$, for each node $v \in R$, we update the marginal gain of the candidate edges pointing to v by subtracting $p_{u,v} p_{u^*,v^*} \pi(R)$. Then we need to update the utility value of each $R \in \mathcal{R}_{v^*}$ by

$$\pi(R) \leftarrow \pi(R)(1 - p_{u^*,v^*}). \tag{12}$$

Repeating such a procedure for k times gives the final solution.

Low probability edge pruning for upper bound maximization. For the upper bounding function, a convenient property is that given two edges $e_1 = \langle u_1, v\rangle$ and $e_2 = \langle u_2, v\rangle$ with $p_{u_1,v} \geq p_{u_2,v}$, we have $\Gamma(A \cup \{e_1\}) \geq \Gamma(A \cup \{e_2\})$. Hence, if e_1 is not yet selected, we do not need to update the marginal gain of e_2. Specifically, we first pack the edges pointing to the same node into an array and sort them according to the edge probability and store the largest marginal gain among the candidate edges pointing to v as $\Delta(v)$. After selecting $\langle u^*, v^*\rangle$, we update $\Delta(v)$ for each node v in every RR set $R \in \mathcal{R}_{v^*}$. In this way, we only maintain the marginal gain of nodes instead of edges.

Putting it together. Above all, the greedy selection algorithms are given in Algorithm 3 and Algorithm 4, which basically follow the same paradigm. In each iteration, we select the edge with the maximum marginal gain and then update others. The key distinction lies in that we we update the gains for nodes but not edges. The time complexity results of them are given by Lemma 5.6.

Lemma 5.6. *Algorithm 3 returns a $(1 - 1/\mathrm{e})$-approximate solution to $\Gamma^L(\cdot)$ in $O(k\,|S| \sum_{R \in \mathcal{R}} |R|)$ time and Algorithm 4 returns a $(1 - 1/\mathrm{e})$-approximate solution to $\Gamma^U(\cdot)$ in $O(k \sum_{R \in \mathcal{R}} |R|)$ time.*

5.4 Sampling Schedule Details

Although the random variable attached with each RR set is no longer a Bernoulli random variable, it remains bounded in $[0, 1]$. Hence, the martingale inequalities underpinning OPIM-C [31] retain applicability. However, some necessary modifications need to be performed to ensure the theoretical guarantee, which are detailed below.

Empirical approximation ratio. Given the analogous forms of the bounds for $\rho^L(\cdot)$ and $\rho^U(\cdot)$, here we focus on $\rho^L(\cdot)$ for conciseness. For any selected A, we construct the lower bound of $\rho^L(A)$ as

$$\rho_l^L(A) = \left(\left(\sqrt{\Gamma_2^L(A) + \frac{2\zeta}{9}} - \sqrt{\frac{\zeta}{2}}\right)^2 - \frac{\zeta}{18}\right) \cdot \frac{n}{\theta_2}, \tag{13}$$

and the upper bound of the optimal solution is

$$\rho_u^L(A^\circ) = \left(\left(\sqrt{\frac{\Gamma_1^L(A)}{1 - 1/\mathrm{e}} + \frac{\zeta}{2}} + \sqrt{\frac{\zeta}{2}}\right)^2\right) \cdot \frac{n}{\theta_1}, \tag{14}$$

where ζ's value is related to failure probability (detailed settings will be revealed later). Then the empirical approximation ratio could be computed by $\rho_l^L(A)/\rho_u^L(A^\circ)$.

Computing $\theta_{\max}$. To determine the worst-case number of samples, we first introduce the theoretical sample size to achieve the guarantee through Lemma 5.7.

Lemma 5.7 (Adapt from [33]). *Let OPT_L be the optimal value of $\rho^L(A)$. Alg. 3 (resp. Alg. 4) returns a $(1 - 1/\mathrm{e} - \varepsilon)$-approximation on $\rho^L(\cdot)$ (resp. $\rho^U(\cdot)$) with probability $1 - \delta/9$ if the number of RR sets θ satisfies*

$$\theta \geq \frac{2n\left((1 - 1/\mathrm{e})\sqrt{\log \frac{18}{\delta}} + \sqrt{(1 - 1/\mathrm{e})(\log \binom{n}{k} + \log \frac{18}{\delta})}\right)^2}{\varepsilon^2 \mathrm{OPT}_L}.$$

As OPT_L is intractable, we need to design a computationally easy way to identify the minimum possible value for OPT_L so as to obtain $\theta_{\max}$. Unlike conventional IM where the minimum possible influence is trivially k, such a lower bound depends on the problem instance. For example, consider all nodes in the network can be influenced with probability 1, indicating that inserting edges will not augment the influence. Therefore, we should first verify whether OPT_L is 0.

Without loss of generality, we suppose that the candidate edge set is not empty, since the problem is meaningless if there are no edge could be inserted into the network. $\mathrm{OPT}_L = 0$ indicates that no candidate edge is able to augment the influence of the given configuration $\mathbf{p}$. That is, the activation probability of each node

Table 2: Datasets ($K = 10^3$, $M = 10^6$, $B = 10^9$)

Name	n	m	Type	Avg. Degree
Epinions	75.9K	508.8K	Directed	13.4
Stanford	281K	2.3M	Directed	16.4
DBLP	317K	1.05M	Undirected	6.1
Orkut	3.1M	117.1M	Undirected	76.3
LiveJournal	4.8M	69M	Directed	28.5
Twitter	41.7M	1.5B	Directed	70.5

cannot be amplified by the candidate edges. This occurs if every node v could be reached from a deterministic seed through a path whose edges' probabilities are all 1, or if there is no candidate edge that points to v. So before running our algorithm, we should first identify the deterministic seeds and run a BFS from them to check if adding edges can increase the influence or not. If $\mathrm{OPT}_L > 0$ is assured, we proceed to compute the lower bound of it.

Here we only consider the common case[1] where there exists at least one node having all incoming edges attached with influence probabilities smaller than 1. For any node v whose activation probability could be augmented, the minimum probability that this node will not be activated by others is $\prod_{u\in N_{in}(v)}(1-p_{u,v})$. By excluding the probability of v itself being an active seed and following the rule of the lower bounding function, the benefit of recommending a seed to v is at least

$$\kappa_v = \max_{s\in S\setminus N_{in}(v)\setminus\{v\}} \mathbf{p}_s(1-\mathbf{p}_v)p_{s,v} \cdot \prod_{u\in N_{in}(v)}(1-p_{u,v}). \tag{15}$$

By summing the top-k largest κ_v values across all $v \in V$, we obtain the minimum possible value of OPT_L, denoted by κ. Then the worst-case sampling size $\theta_{\max}$ is defined as

$$\theta_{\max} = \frac{2n\left((1-1/\mathrm{e})\sqrt{\log\frac{18}{\delta}} + \sqrt{(1-1/\mathrm{e})(\log\binom{n}{k} + \log\frac{18}{\delta})}\right)^2}{\varepsilon^2\kappa}. \tag{16}$$

5.5 Putting It Together

Now we put it together and give theoretical analysis on the JB-PIUS. Specifically, in the implementation of Algorithm 2, we set $\theta_{\max}$ by Equation (16), and compute the empirical approximation ratio by Equation (13) and (14) with $\zeta = \frac{2\delta}{9i_{\max}}$. Finally, for the evaluation of A_L, A_U and A_H, we conduct $(\gamma, \delta/9)$-estimation on the augmented influence spread by utilizing the general stopping rules [44]. Combining these components, the approximation guarantee is established by the following theorem.

Theorem 5.8. *The solution A^* returned by Algorithm 2 satisfies*

$$\sigma(A^*,\mathbf{p}) \geq \max\left\{\frac{\sigma^L(A_L^\circ,\mathbf{p})}{\sigma(A^\circ,\mathbf{p})}, \frac{\sigma(A_U,\mathbf{p})}{\sigma^U(A_U,\mathbf{p})}\right\}\frac{1-\gamma}{1+\gamma}(1-1/\mathrm{e}-\varepsilon)\sigma(A^\circ,\mathbf{p})$$

with probability at least $1-\delta$.

The time complexity of JB-PIUS consists of RR sets sampling, selection and the final estimation stages. Combining the analysis of existing work [31] and Lemma 5.6, we have the following theorem regarding the running time of Algorithm 2.

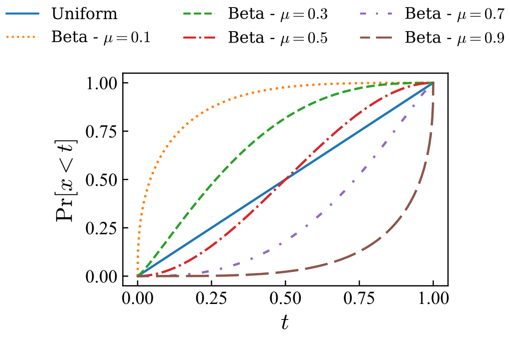


Figure 4: Cumulative distribution functions of all tested seed probability distributions.

Table 3: Parameters settings in our experiments.

Name	Default	Values
k	100	$\{5, 10, 20, 40, 60, 80, 100\}$
$\lvert S\rvert$	50	$\{10, 20, 30, 40, 50\}$
Seed Dist.	Uniform	{Uniform, Beta}
μ	-	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$
ε	0.1	$\{0.1, 0.2, 0.3, 0.4, 0.5\}$
γ	0.05	$\{0.05, 0.1, 0.15\}$

Theorem 5.9. *Algorithm 2 runs in* $O(\frac{(k\ln n+\ln(1/\delta))n\mathrm{EPT}}{\varepsilon^2\mathrm{OPT}_L} + k\,|S| \cdot \sum_{R\in\mathcal{R}}|R| + \frac{n\ln(1/\delta)}{\gamma^2}\cdot\frac{\mathrm{EPT}}{\sigma(A^*,\mathbf{p})})$ *time, where* EPT *denotes the expected time for sampling a random RR set.*

6 EXPERIMENTS

6.1 Experiment Settings

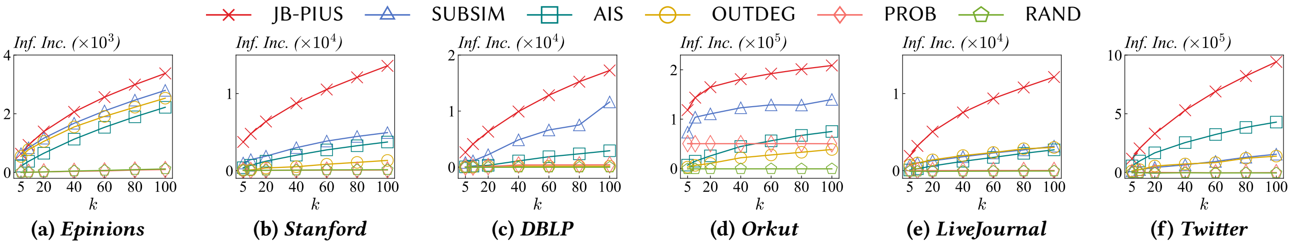
This section evaluates the empirical performance of the proposed algorithm. All experiments are conducted on a linux machine with Intel Xeon(R) 8377C@3.0GHz and 512GB RAM. All algorithms are implemented by C++ with O3 optimization. In our experiments, we run each method for 10 times and report the average results.

Datasets. We conduct experiments on 6 publicly available datasets [25, 27], the statistics of which are presented in Table 2. Among them, Twitter is one of the largest dataset used in the field of influence analysis with 1.5 billion edges.

Parameters. The propagation probability $p_{u,v}$ of each edge is set to be the inverse of v's in-degree, which is a widely-adopted setting [5, 19, 22, 31, 33, 34]. The failure probability δ of our algorithm is fixed as $1/n$. Under the default setting, we set the seed set size $|S| = 50$, with the seed nodes selected by state-of-the-art IM algorithm [16], conforming the real scenario where we will select influential nodes as seed users. Other default parameters include $k = 100$, $\varepsilon = 0.1$ and $\gamma = 0.05$. For the configuration $\mathbf{p}$, we assign a uniform random number in $(0, 1]$ to $\mathbf{p}_s$ for each $s \in S$. When studying the effect of different seeding probabilities, we employ Beta distribution with varying means μ for the probability assignment, whose cumulative distribution curves are shown in Figure 4. All parameter settings tested are summarized in Table 3.

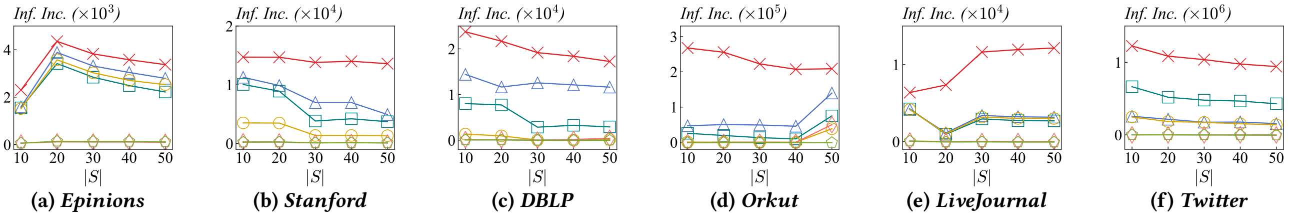
To demonstrate the scalability of the proposed algorithm, the candidate edge set E_C contains all possible edges, which means that

[1] A graph where every node has at least one in-neighbor with influence probability 1 can be converted to a DAG by Tarjan's algorithm. We can erase the loops and find a node whose incoming edges are with influence probability smaller than 1.



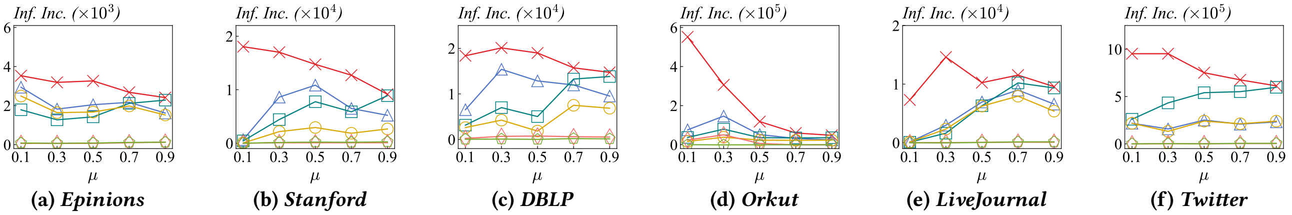
(a) ***Epinions*** (b) ***Stanford*** (c) ***DBLP*** (d) ***Orkut*** (e) ***LiveJournal*** (f) ***Twitter***

Figure 5: Influence increment with varying k.



(a) ***Epinions*** (b) ***Stanford*** (c) ***DBLP*** (d) ***Orkut*** (e) ***LiveJournal*** (f) ***Twitter***

Figure 6: Influence increment with varying $|S|$.



(a) ***Epinions*** (b) ***Stanford*** (c) ***DBLP*** (d) ***Orkut*** (e) ***LiveJournal*** (f) ***Twitter***

Figure 7: Influence increment with varying μ.

the size of E_C is $\Theta(|S|\,n)$. Every candidate edge $e = \langle u, v\rangle$ contains a seed node $u \in S$ and another node $v \in V \setminus \{u\}$. Taking into account both the probability of u activating its out-neighbors and the probability of v being activated [8], the propagation probability of a candidate edge $\langle u, v\rangle$ is assigned as $\frac{\bar{p}_u^{\text{out}}+\bar{p}_v^{\text{in}}}{2}$, where $\bar{p}_u^{\text{out}} = \frac{\sum_{w\in N_{\text{out}}(u)} p_{u,w}}{|N_{\text{out}}(u)|}$ and $\bar{p}_v^{\text{in}} = \frac{\sum_{w\in N_{\text{in}}(v)} p_{w,v}}{|N_{\text{in}}(v)|}$.

Algorithms. We compare JB-PIUS with the following methods.

- SUBSIM [20]. The state-of-the-art IM algorithm.
- AIS [8]. An RIS-based method for IMA [9, 10]. The number of RR sets is set as the requirement to obtain a (γ, δ)-estimation on $\sigma(\mathbf{p})$.
- OUTDEG. Select the edges pointing to top-k nodes with the highest out-degree.
- PROB. Select top-k edges with the highest influence probability.
- RAND. Select k edges uniformly at random.

Evaluation metrics. To evaluate the effectiveness and efficiency of the proposed algorithm, we mainly investigate the following three metrics in our experiments.

- *Influence Increment.* The influence increment is computed by $\sigma(A, \mathbf{p}) - \sigma(\mathbf{p})$, where the influence spread is estimated by RIS [40] within a multiplicative error 0.01.
- *Running Time.* The running time of each algorithm is recorded for the efficiency test, excluding I/O of the graph.
- *Approximation Ratio.* Following existing work [37], we compute $\left(\frac{1-\gamma}{1+\gamma}\right)^2 \cdot (1 - 1/\mathrm{e} - \varepsilon) \cdot \frac{\hat{\sigma}(A_U, \mathbf{p})}{\hat{\sigma}^U(A_U, \mathbf{p})}$ as the lower bound of the data-dependent approximation ratio, which reflects the quality of the approximation.

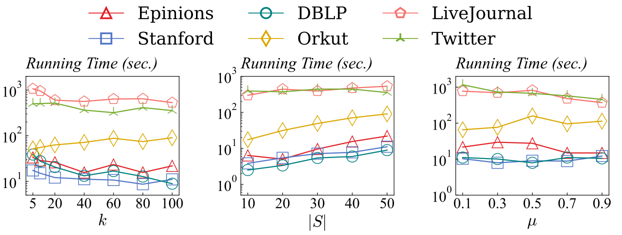
6.2 Effectiveness Evaluation

Varying k. Figure 5 reports the resulting influence increment with varying k on all six datasets. It can be observed that JB-PIUS consistently outperforms the baselines in terms of influence increment, demonstrating JB-PIUS's effectiveness. Naive heuristics (i.e., OUTDEG, PROB and RAND) always fail to provide effective solutions to IMAUS with varying k. SUBSIM and AIS surpass heuristic approaches by targeting influential nodes, yet remain suboptimal compared to JB-PIUS. This highlights JB-PIUS 's superior ability to identify profitable edges to augment the influence.

Varying $|S|$. Figure 6 compares the influence increment of all methods w.r.t. different number of seeds $|S|$. The results indicate that JB-PIUS maintains consistent superiority to other baselines. An interesting observation is that the influence increment decreases with larger $|S|$ in some cases. This phenomenon arises because when more influential nodes are selected as seeds, recommending links

Table 4: Effect of bounding functions maximization with varying μ

	Epinions		Stanford		DBLP		Orkut		LiveJournal		Twitter	
k	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
$\mu = 0.1$	**3521.886**	3491.871	**18026.23**	1035.405	**18361.12**	5659.7	**550891.3**	259652.2	**7348.29**	1810.762	**949854**	436873
$\mu = 0.3$	**3201.789**	2294.038	**17038.83**	12239.65	**19989.97**	12789.7	**305012.4**	272334.4	**14666.89**	10937.19	**951837**	583227
$\mu = 0.5$	**3233.06**	2869.562	**14725.63**	13906.85	**19042.2**	7287.94	**118293.5**	113809.8	**10379.97**	9444.46	**782688**	650558
$\mu = 0.7$	2687.01	**2688.48**	**12756.27**	9953	**15721.44**	15007.25	58673.6	**59272**	11491.35	**11665.1**	**681954**	572624
$\mu = 0.9$	**2442.13**	2385.09	9084.32	**9187.32**	14665.55	**14838.35**	**48497.8**	47169.9	**10019.56**	9385.2	592525	**620029**



(a) *Varying* k **(b) *Varying* $|S|$** **(c) *Varying* μ**

Figure 8: Running time for different parameters.

to influential nodes will have a lower benefit given that such nodes may already be selected as seeds. But some datasets like LiveJournal show an upward trend. A possible reason is that larger $|S|$ expands candidate edge set, which may include more high-impact edges.

Varying seed probability distribution. To evaluate robustness under different seed activation probability distributions, we initialize seeding probabilities via Beta distributions with varying means μ. As presented in Figure 7, we can observe that JB-PIUS consistently achieves superior influence increment over other methods. Notably, when μ is large enough, AIS tends to have similar performance with JB-PIUS, since when the seeds are close to being deterministically activated, IMAUS will become IMA and AIS provides satisfactory results. In addition, we observe that the influence increment tends to decrease for most datasets when μ is large since high activation probabilities leave fewer nodes available for influence augmentation.

Effectiveness of the bounding functions. We investigate the resulting influence increment of the solutions returned by lower bound (LB) and upper bound (UB) maximization algorithms, respectively. Table 4 shows the resulting influence increment of LB and UB with varying μ. We can see that the LB maximization algorithm provides more satisfactory solutions than UB in most scenarios, indicating that the proposed LB is tight. Notably, the quality of UB's solution quality improves with higher μ, aligning with its design assumption that inserted edges originate from deterministically activated seeds - a condition increasingly met as μ approaches 1.

6.3 Efficiency Evaluation

Varying k, $|S|$ and μ. Figure 8 reports the algorithm's runtime across varying k, $|S|$ and μ. For datasets except LiveJournal and Twitter, it only takes less than two minutes for our algorithm to finish. For LiveJournal and Twitter, our algorithm is able to finish within 20 minutes under all settings. When increasing k, the running time remains stable for all datasets. We can see that the running time increases with larger $|S|$ since the candidate edge set is also enlarged, thereby increasing selection time. As for the efficiency with different μ, it can be observed that the algorithm's time cost does not present obvious trends with larger μ.

Table 5: Running time (sec.) breakdown of JB-PIUS

Procedure	Epinions	Stanford	DBLP	Orkut	LiveJournal	Twitter
Sampling	0.684	1.128	0.671	1.473	28.968	7.854
Selection	20.641	8.857	7.811	87.870	471.840	244.690
Estimation	0.585	1.106	0.436	1.186	30.945	103.046

Running time breakdown. Table 5 details the runtime breakdown of JB-PIUS, which completes execution in under 10 minutes even for the largest two datasets. The time cost is dominated by the selection phase since the search space is $\Theta(|S|\,n)$, unlike $O(n)$ for IM. The time cost for the solution quality estimation is at the same order of magnitude compared with that of the sampling phase for all datasets except Twitter.

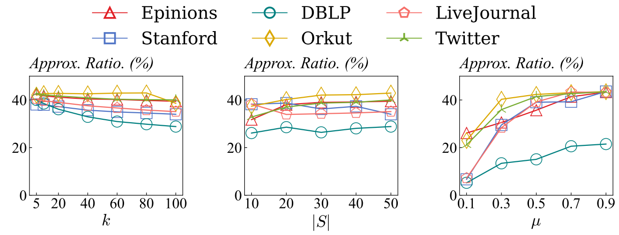
6.4 Approximation Quality Evaluation

We compute the lower bound of data-dependent approximation ratio for each dataset with different values of k, $|S|$ and μ to evaluate the approximation quality. The results are illustrated in Figure 9.

Varying k. Figure 9a show that JB-PIUS achieves the highest empirical approximation ratio of over 40% on Orkut. When $k = 5$, the approximation ratio is around 40% for all datasets. With larger k, the empirical guarantee of different datasets decreases at different rates. The reason may be that the submodular upper bound becomes looser in that case. Notably, sparser networks (e.g., DBLP) exhibit sharper declines, with the empirical ratio falling to less than 30% when $k = 100$.

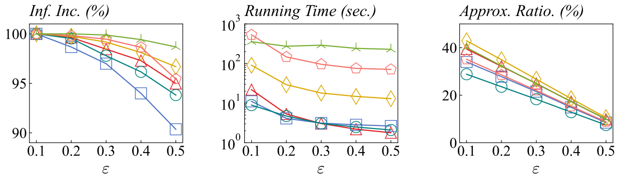
Varying $|S|$. Figure 9b reveals that the empirical approximation ratio of the proposed algorithm remains stable across varying $|S|$, which suggests that including more seeds does not affect the approximation quality of the proposed algorithms.

Varying μ. Figure 9c demonstrates that the empirical approximation ratio increases with higher μ, which implies that better approximations could be obtained when the activation probability of the seeds is high. An important factor in the computation of the ratio is $\sigma(A_U, \mathbf{p})/\sigma^U(A_U, \mathbf{p})$. Recall that the upper bounding function assumes that each candidate edge points from a deterministic seed. Therefore, when μ increases, the seed nodes have higher probability to be activated, and $\sigma^U(A_U, \mathbf{p})$ is closer to $\sigma(A_U, \mathbf{p})$, thereby increasing the empirical approximation ratio.



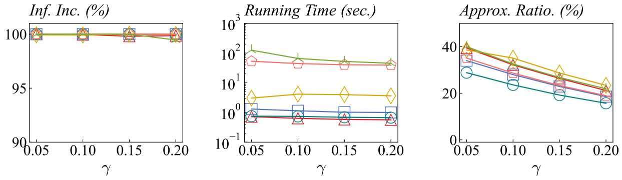
(a) ***Varying*** k (b) ***Varying*** $|S|$ (c) ***Varying*** μ

Figure 9: Approximation ratio with varying μ on all datasets.



(a) ***Influence increment*** (b) ***Running time*** (c) ***Approx. Ratio***

Figure 10: Varying ε.



(a) ***Influence increment*** (b) ***Running time*** (c) ***Approx. Ratio***

Figure 11: Varying γ.

6.5 Sensitivity Evaluation

Now we turn to the sensitivity evaluation of the proposed algorithm w.r.t. different ε and γ.

Effect of ε. We vary ε from 0.1 to 0.5 to evaluate the influence increment, running time and approximation ratio, respectively. As shown in Figure 10, with the increase of ε, the resulting influence increment decreases for all datasets since the approximation ratios of the bounding functions decreases. Despite this, when $\varepsilon = 0.5$, JB-PIUS maintains high-quality solutions with less than 10% compromise. Since larger ε leads to earlier stopping for the sampling phase, the running time of our algorithm decreases accordingly. Figure 10c confirms a linear decline in approximation ratios with ε we can see that the approximation ratio decreases linearly with the increase of ε as ε directly scales the data-dependent approximation ratio. The results trade-off underscore JB-PIUS 's ability to balance efficiency and precision through tunable ε.

Effect of γ. Now we evaluate our method's sensitivity to γ, the results of which are given in Figure 11. Since the parameter γ solely governs the final selection among A_L, A_U and A_H, the resulting influence increment exhibits minimal variation with increasing γ. As for efficiency, runtime improvements remain marginal across most datasets, as the estimation phase contributes negligibly to total execution time. On Twitter, however, estimation takes considerable time, rendering runtime improvements more obvious. The approximation ratio declines due to the $\frac{1+\gamma}{1-\gamma}$ term, which directly scales the data-dependent ratio.

7 ADDITIONAL RELATED WORK

Influence maximization. Domingos and Richardson [11] are pioneers to take advantage of the word-of-mouth effect to enhance purchasing behavior within social networks. Then, Kempe et al. [22] formalize the well-known influence maximization (IM) problem. They show that this problem is NP-hard, and propose a simulation-based greedy algorithm to approximate the solution. Subsequently, a plethora of research works study how to improve the efficiency to tackle the problem [2, 5–7, 13, 14, 21, 26, 30], most of which are heuristic and fail to provide approximation guarantees. Later, Borgs et al. [3] achieve a notable advancement by introducing an algorithm that runs in near-linear time under the IC model, maintaining a $(1 - 1/e - \varepsilon)$-approximation ratio. They present the concept of reverse influence sampling (RIS) and convert the IM problem into a maximum coverage problem with this technique. Their work underpins the development of state-of-the-art solutions for IM [16, 19, 29, 31, 33, 34]. Different from the discrete scenario, Yang et al. [40, 41] study continuous IM, which aims to offer different discounts to customers with a limited budget such that the resulting influence is maximized.

Topology optimization for IM. In addition to the focus on identifying influential nodes, the subject of enhancing network connectivity to amplify influence propagation has also garnered significant interest from researchers [4, 9, 10, 20, 23, 24, 38, 39]. This line of research aims to maximize the influence spread of a predetermined seed set by inserting a limited number of edges into the social network. Although the objectives are roughly similar, the settings of existing work are various. To name a few, Chaoji et al. [4] study the problem of adding k edges for each node to maximize the influence under a proxy model for IC. Khalil et al. [23] consider adding edges to maximize influence spread under the LT model, but the objective function is the sum of influence spread of each seed. The IMA problem proposed by D'Angelo et al. [10] aims to select k edges incident to the seed set to maximize the influence of the seed set S. The objective function is monotone and submodular [10], and several greedy-based algorithms [8, 9] have been proposed to achieve an approximation ratio of $(1 - 1/e - \varepsilon)$ [28] for IMA. All of the aforementioned work assumes deterministic seeds, which presents limited generalizability for the probabilistic seeding scenarios.

8 CONCLUSION

In this paper, we study the problem of inserting links to promote the influence of uncertain seeds in social networks. We first formulate it as an NP-hard combinatorial optimization problem and prove that the objective function is monotone but neither submodular nor supermodular. To provide a solution with certain theoretical guarantees, we instantiate the sandwich approximation strategy with two newly proposed monotone and submodular bounding functions. For the detailed implementation, we propose a joint baking algorithm named JB-PIUS consisting of two efficient estimators for the bounding functions as well as some well-designed greedy selection algorithms. Finally, we conduct extensive experiments to validate our method's effectiveness and efficiency, whose results show that JB-PIUS provides higher influence spread than other competitors and scales well to million-size datasets.

REFERENCES

[1] Florian Adriaens, Honglian Wang, and Aristides Gionis. 2023. Minimizing Hitting Time between Disparate Groups with Shortcut Edges. *arXiv preprint arXiv:2306.03571* (2023).

[2] Shishir Bharathi, David Kempe, and Mahyar Salek. 2007. Competitive influence maximization in social networks. In *International workshop on web and internet economics*. Springer, 306–311.

[3] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 946–957.

[4] Vineet Chaoji, Sayan Ranu, Rajeev Rastogi, and Rushi Bhatt. 2012. Recommendations to boost content spread in social networks. In *Proceedings of the 21st international conference on World Wide Web*. 529–538.

[5] Wei Chen, Chi Wang, and Yajun Wang. 2010. 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*. 1029–1038.

[6] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 199–208.

[7] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *2010 IEEE international conference on data mining*. IEEE, 88–97.

[8] Xiaolong Chen, Yifan Song, and Jing Tang. 2024. Link Recommendation to Augment Influence Diffusion with Provable Guarantees. In *Proceedings of the ACM on Web Conference 2024*. 2509–2518.

[9] Federico Coró, Gianlorenzo D'angelo, and Yllka Velaj. 2021. Link recommendation for social influence maximization. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 6 (2021), 1–23.

[10] Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. 2019. Recommending links through influence maximization. *Theoretical Computer Science* 764 (2019), 30–41.

[11] Pedro Domingos and Matt Richardson. 2001. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. 57–66.

[12] Ding-Zhu Du, Panos M Pardalos, Xiaodong Hu, and Weili Wu. 2022. *Introduction to combinatorial optimization*. Vol. 196. Springer Nature.

[13] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. 2011. Celf++ optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*. 47–48.

[14] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. 2011. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *2011 IEEE 11th international conference on data mining*. IEEE, 211–220.

[15] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A Zighed. 2013. Information diffusion in online social networks: A survey. *ACM Sigmod Record* 42, 2 (2013), 17–28.

[16] Qintian Guo, Sibo Wang, Zhewei Wei, Wenqing Lin, and Jing Tang. 2022. Influence Maximization Revisited: Efficient Sampling with Bound Tightened. *ACM Transactions on Database Systems (TODS)* 47, 3 (2022), 1–45.

[17] Shahrzad Haddadan, Cristina Menghini, Matteo Riondato, and Eli Upfal. 2021. Repbublik: Reducing polarized bubble radius with link insertions. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 139–147.

[18] Zheng Hu, Weiguo Zheng, and Xiang Lian. 2023. Triangular stability maximization by influence spread over social networks. *Proceedings of the VLDB Endowment* 16, 11 (2023), 2818–2831.

[19] Keke Huang, Sibo Wang, Glenn Bevilacqua, Xiaokui Xiao, and Laks VS Lakshmanan. 2017. Revisiting the stop-and-stare algorithms for influence maximization. *Proceedings of the VLDB Endowment* 10, 9 (2017), 913–924.

[20] Shixun Huang, Wenqing Lin, Zhifeng Bao, and Jiachen Sun. 2022. Influence maximization in real-world closed social networks. *Proceedings of the VLDB Endowment* 16, 2 (2022), 180–192.

[21] Kyomin Jung, Wooram Heo, and Wei Chen. 2012. Irie: Scalable and robust influence maximization in social networks. In *2012 IEEE 12th international conference on data mining*. IEEE, 918–923.

[22] David Kempe, Jon Kleinberg, and Éva Tardos. 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.

[23] Elias Boutros Khalil, Bistra Dilkina, and Le Song. 2014. Scalable diffusion-aware optimization of network topology. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1226–1235.

[24] Chris J Kuhlman, Gaurav Tuli, Samarth Swarup, Madhav V Marathe, and SS Ravi. 2013. Blocking simple and complex contagion by edge removal. In *2013 IEEE 13th international conference on data mining*. IEEE, 399–408.

[25] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a social network or a news media?. In *WWW '10: Proceedings of the 19th international conference on World wide web* (Raleigh, North Carolina, USA). ACM, New York, NY, USA, 591–600. https://doi.org/10.1145/1772690.1772751

[26] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 420–429.

[27] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[28] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming* 14 (1978), 265–294.

[29] Hung T Nguyen, My T Thai, and Thang N Dinh. 2016. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *Proceedings of the 2016 international conference on management of data*. 695–710.

[30] Naoto Ohsaka, Takuya Akiba, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2014. Fast and accurate influence maximization on large networks with pruned monte-carlo simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[31] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online processing algorithms for influence maximization. In *Proceedings of the 2018 International Conference on Management of Data*. 991–1005.

[32] Jing Tang, Xueyan Tang, and Junsong Yuan. 2017. Influence maximization meets efficiency and effectiveness: A hop-based approach. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. 64–71.

[33] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1539–1554.

[34] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 75–86.

[35] Guangmo Tong, Weili Wu, Shaojie Tang, and Ding-Zhu Du. 2016. Adaptive influence maximization in dynamic social networks. *IEEE/ACM Transactions on Networking* 25, 1 (2016), 112–125.

[36] Sotiris Tsioutsiouliklis, Evaggelia Pitoura, Konstantinos Semertzidis, and Panayiotis Tsaparas. 2022. Link Recommendations for PageRank Fairness. In *Proceedings of the ACM Web Conference 2022*. 3541–3551.

[37] Zhefeng Wang, Yu Yang, Jian Pei, Lingyang Chu, and Enhong Chen. 2017. Activity Maximization by Effective Information Diffusion in Social Networks. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2374–2387. https://doi.org/10.1109/TKDE.2017.2740284

[38] Wenguo Yang, Shengminjie Chen, Suixiang Gao, and Ruidong Yan. 2020. Boosting node activity by recommendations in social networks. *Journal of Combinatorial Optimization* 40, 3 (2020), 825–847.

[39] Wenguo Yang, Jianmin Ma, Yi Li, Ruidong Yan, Jing Yuan, Weili Wu, and Deying Li. 2019. Marginal gains to maximize content spread in social networks. *IEEE Transactions on Computational Social Systems* 6, 3 (2019), 479–490.

[40] Yu Yang, Xiangbo Mao, Jian Pei, and Xiaofei He. 2016. Continuous influence maximization: What discounts should we offer to social network users?. In *Proceedings of the 2016 international conference on management of data*. 727–741.

[41] Yu Yang, Xiangbo Mao, Jian Pei, and Xiaofei He. 2020. Continuous influence maximization. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14, 3 (2020), 1–38.

[42] Shiqi Zhang, Yiqian Huang, Jiachen Sun, Wenqing Lin, Xiaokui Xiao, and Bo Tang. 2023. Capacity Constrained Influence Maximization in Social Networks. *arXiv preprint arXiv:2306.01782* (2023).

[43] Liwang Zhu, Qi Bao, and Zhongzhi Zhang. 2021. Minimizing polarization and disagreement in social networks via link recommendation. *Advances in Neural Information Processing Systems* 34 (2021), 2072–2084.

[44] Yuqing Zhu, Jing Tang, Xueyan Tang, Sibo Wang, and Andrew Lim. 2021. 2-hop+ sampling: Efficient and effective influence estimation. *IEEE Transactions on Knowledge and Data Engineering* (2021).

A MISSING PROOFS

Proof of Lemma 5.3. According to the definition of $\Gamma^L(\cdot)$,

$$\Gamma^L(A \cup \{x, y\}) = \sum_{R \in \mathcal{R}_x} \pi_0(R)(1 - \prod_{u \in S \backslash R} (1 - \mathbf{p}_u w(u, R, A)))$$
$$+ \sum_{R \in \mathcal{R} \backslash \mathcal{R}_x} \pi_0(R)(1 - \prod_{u \in S \backslash R \backslash \{x\}} (1 - \mathbf{p}_u w(u, R, A))(1 - \mathbf{p}_x w(x, R, A \cup \{\langle x, y \rangle\}))).$$

Then

$$\begin{aligned}\Gamma^L(A \cup \{\langle x, y \rangle\}) - \Gamma^L(A) &= \sum_{R \in \mathcal{R} \backslash \mathcal{R}_x} \pi_0(R) \cdot \prod_{u \in S \backslash R \backslash \{x\}} (1 - \mathbf{p}_u w(u, R, A)) \cdot \mathbf{p}_x (w(x, R, A \cup \{\langle x, y \rangle\}) - w(x, R, A)) \\ &= \sum_{R \in \mathcal{R} \backslash \mathcal{R}_x} \pi_0(R) \cdot \prod_{u \in S \backslash R \backslash \{x\}} (1 - \mathbf{p}_u w(u, R, A)) \cdot \mathbf{p}_x p_{x,y} \mathbb{I}(y \in R) \prod_{\langle x, v \rangle \in A_x^+} (1 - p_{x,v} \mathbb{I}(v \in R)) \\ &= \sum_{R \in \mathcal{R} \backslash \mathcal{R}_x} \pi_0(R) \cdot \prod_{u \in S \backslash R \backslash \{x\}} (1 - \mathbf{p}_u w(u, R, A)) \cdot \mathbf{p}_x p_{x,y} \mathbb{I}(y \in R)(1 - w(x, R, A)) \geq 0.\end{aligned}$$

So $\Gamma^L(\cdot)$ is monotone non-decreasing. Similarly, for any $B \supseteq A$, we have

$$\Gamma^L(B \cup \{\langle x, y \rangle\}) - \Gamma^L(B) = \sum_{R \in \mathcal{R} \backslash \mathcal{R}_x} \pi_0(R) \cdot \prod_{u \in S \backslash R \backslash \{x\}} (1 - \mathbf{p}_u w(u, R, B)) \cdot \mathbf{p}_x p_{x,y} (1 - w(x, R, B)).$$

As $w(u, R, A) \leq w(u, R, B)$ if $A \subseteq B$, We naturally have $\Gamma^L(B \cup \{\langle x, y \rangle\}) - \Gamma^L(B) \leq \Gamma^L(A \cup \{\langle x, y \rangle\}) - \Gamma^L(A)$. So the submodularity holds.

Then we turn to prove the monotonicity and submodularity of $\Gamma^U(\cdot)$.

$$\Gamma^U(A \cup \{\langle x, y \rangle\}) = \sum_{R \in \mathcal{R} \backslash \mathcal{R}_y} \pi_0(R)(1 - \prod_{v \in R} \prod_{\langle u, v \rangle \in A_v^-} (1 - p_{u,v})) + \sum_{R \in \mathcal{R}_y} \pi_0(R)(1 - \prod_{v \in R} \prod_{\langle u, v \rangle \in A_v^-} (1 - p_{u,v}) \cdot (1 - p_{x,y})).$$

Then

$$\Gamma^U(A \cup \{\langle x, y \rangle\}) - \Gamma^U(A) = \sum_{R \in \mathcal{R}_y} \pi_0(R) p_{x,y} \cdot \prod_{v \in R} \prod_{\langle u, v \rangle \in A_v^-} (1 - p_{u,v}) > 0,$$

which proves the monotonicity of $\Gamma^U(\cdot)$. For any $B \supseteq A$,

$$\Gamma^U(B \cup \{\langle x, y \rangle\}) - \Gamma^U(B) = \sum_{R \in \mathcal{R}_y} \pi_0(R) p_{x,y} \cdot \prod_{v \in R} \prod_{\langle u, v \rangle \in A_v^-} (1 - p_{u,v}) \prod_{\langle u, v \rangle \in B_v^- \backslash A_v^-} (1 - p_{u,v})$$

Hence $\Gamma^U(B \cup \{\langle x, y \rangle\}) - \Gamma^U(B) \leq \Gamma^U(A \cup \{\langle x, y \rangle\}) - \Gamma^U(A)$, which finishes the proof. □

Proof of Lemma 5.4.

$$\begin{aligned}\Gamma^L(\langle u', v' \rangle \mid A) &= \sum_{R \in \mathcal{R}} \pi_0(R) \mathbb{I}(u' \notin R) \prod_{S \backslash R \backslash \{u'\}} (1 - \mathbf{p}_u w(w, R, A)) \mathbf{p}_{u'} \cdot (w(u', R, A \cup \{\langle u', v' \rangle\}) - w(u', R, A)) \\ &= \sum_{R \in \mathcal{R} \backslash \mathcal{R}_{u'}} \pi_0(R) \prod_{S \backslash R \backslash \{u'\}} (1 - \mathbf{p}_u w(w, R, A)) \mathbf{p}_{u'} \cdot \prod_{\langle u', v \rangle \in A_{u'}^+} (1 - p_{u',v} \mathbb{I}(v \in R)) p_{u',v'} \mathbb{I}(v' \in R) \\ &= \sum_{R \in \mathcal{R}_{v'} \backslash \mathcal{R}_{u'}} \pi_0(R) \prod_{S \backslash R \backslash \{u'\}} (1 - \mathbf{p}_u w(u, R, A)) \mathbf{p}_{u'} p_{u',v'} (1 - w(u', R, A))\end{aligned}$$

And for the marginal gain of $\langle u', v' \rangle$ after selecting $\langle u^*, v^* \rangle$, we discuss two cases: (i) $u' = u^*$ and (ii) $u' \neq u^*$.

$$\Gamma^L(\langle u', v' \rangle \mid A \cup \{\langle u^*, v^* \rangle\}) = \sum_{R \in \mathcal{R}_{v'} \backslash \mathcal{R}_{u'}} \pi_0(R) \prod_{S \backslash R \backslash \{u'\}} (1 - \mathbf{p}_u w(u, R, A)) \mathbf{p}_{u'} \cdot (1 - w(u', R, A \cup \{\langle u^*, v^* \rangle\})) p_{u',v'}$$

So we have

$$\begin{aligned}&\Gamma^L(\langle u', v' \rangle \mid A) - \Gamma^L(\langle u', v' \rangle \mid A \cup \{\langle u^*, v^* \rangle\}) \\ &= \sum_{R \in \mathcal{R}_{v'} \backslash \mathcal{R}_{u'}} \pi_0(R) \prod_{S \backslash R \backslash \{u'\}} (1 - \mathbf{p}_u w(u, R, A)) \mathbf{p}_{u'} p_{u',v'} \cdot (w(u^*, R, A \cup \{\langle u^*, v^* \rangle\}) - w(u^*, R, A)) \\ &= \sum_{R \in \mathcal{R}_{v'} \backslash \mathcal{R}_{u'}} \pi_0(R) \prod_{S \backslash R \backslash \{u'\}} (1 - \mathbf{p}_u w(u, R, A)) \mathbf{p}_{u'} p_{u',v'} \cdot p_{u^*,v^*} \mathbb{I}(v^* \in R)(1 - w(u^*, R, A))\end{aligned}$$

$$= \sum_{R \in \mathcal{R}_{v'} \cap \mathcal{R}_{v^*} \setminus \mathcal{R}_{u'}} \pi(R) \mathbf{p}_{u'} p_{u',v'} p_{u^*,v^*} \frac{1 - w(u^*, R, A)}{1 - \mathbf{p}_{u^*} w(u^*, R, A)}$$

For the case $u' \neq u^*$, we have

$$\Gamma^L(\langle u', v' \rangle \mid A \cup \{\langle u^*, v^* \rangle\}) = \sum_{R \in \mathcal{R}_{v'} \setminus \mathcal{R}_{u'} \setminus \mathcal{R}_{u^*}} \pi_0(R) \prod_{u \in S \setminus R \setminus \{u', u^*\}} (1 - \mathbf{p}_u w(u, R, A)) \cdot (1 - \mathbf{p}_{u^*} w(u, R, A \cup \{ \langle u^*, v^* \rangle\})) \cdot \mathbf{p}_{u'} p_{u',v'} (1 - w(u', R, A))$$
$$+ \sum_{R \in \mathcal{R}_{v'} \setminus \mathcal{R}_{u'} \cap \mathcal{R}_{u^*}} \pi_0(R) \prod_{S \setminus R \setminus \{u'\}} (1 - \mathbf{p}_u w(u, R, A)) \mathbf{p}_{u'} \cdot (1 - w(u', R, A)) p_{u',v'}$$

Then we have

$$\begin{aligned}
&\Gamma^L(\langle u', v' \rangle \mid A) - \Gamma^L(\langle u', v' \rangle \mid A \cup \{\langle u^*, v^* \rangle\}) \\
&= \sum_{R \in \mathcal{R}_{v'} \setminus \mathcal{R}_{u'} \setminus \mathcal{R}_{u^*}} \pi_0(R) \prod_{u \in S \setminus R \setminus \{u', u^*\}} (1 - \mathbf{p}_u w(u, R, A)) \cdot \mathbf{p}_{u'} p_{u',v'} (1 - w(u', R, A)) \cdot \mathbf{p}_{u^*} (w(u^*, R, A \cup \{ \langle u^*, v^* \rangle\}) - w(u^*, R, A)) \\
&= \sum_{R \in \mathcal{R}_{v'} \setminus \mathcal{R}_{u'} \setminus \mathcal{R}_{u^*}} \pi_0(R) \prod_{u \in S \setminus R \setminus \{u', u^*\}} (1 - \mathbf{p}_u w(u, R, A)) \cdot \mathbf{p}_{u'} p_{u',v'} (1 - w(u', R, A)) \cdot \mathbf{p}_{u^*} p_{u^*,v^*} \mathbb{I}(v^* \in R)(1 - w(u^*, R, A)) \\
&= \sum_{R \in \mathcal{R}_{v'} \cap \mathcal{R}_{v^*} \setminus \mathcal{R}_{u'} \setminus \mathcal{R}_{u^*}} \pi(R) \mathbf{p}_{u'} p_{u',v'} \frac{1 - w(u', R, A)}{1 - \mathbf{p}_{u'} w(u', R, A)} \cdot \mathbf{p}_{u^*} p_{u^*,v^*} \frac{1 - w(u^*, R, A)}{1 - \mathbf{p}_{u^*} w(u^*, R, A)}
\end{aligned}$$

The proof is complete. □

Proof of Lemma 5.5. By the definition of $\Gamma^U(\cdot)$,

$$\Gamma^U(\langle u', v' \rangle \mid A) = \sum_{R \in \mathcal{R}} \mathbb{I}(v' \in R) \cdot \pi_0(R) \prod_{v \in R} \prod_{\langle u, v \rangle \in A_v^-} (1 - p_{u,v}) p_{u',v'} = \sum_{R \in \mathcal{R}_{v'}} \pi(R) p_{u',v'}.$$

Similarly,

$$\begin{aligned}
&\Gamma^U(\langle u', v' \rangle \mid A \cup \{\langle u^*, v^* \rangle\}) \\
&= \sum_{R \in \mathcal{R}} \mathbb{I}(v' \in R) \cdot \pi_0(R) \prod_{v \in R \setminus \{v^*\}} \prod_{\langle u, v \rangle \in A_v^-} (1 - p_{u,v}) p_{u',v'} \cdot \prod_{\langle u, v^* \rangle \in A_{v^*}^-} (1 - p_{u,v^*} \mathbb{I}(v^* \in R)) \cdot (1 - p_{u^*,v^*} \mathbb{I}(v^* \in R)) \\
&= \sum_{R \in \mathcal{R}_{v'}} \pi_0(R) \prod_{v \in R} \prod_{\langle u, v \rangle \in A_v^-} (1 - p_{u,v}) p_{u',v'} \cdot (1 - p_{u^*,v^*} \mathbb{I}(v^* \in R)) \\
&= \sum_{R \in \mathcal{R}_{v'}} \pi(R) p_{u',v'} (1 - p_{u^*,v^*} \mathbb{I}(v^* \in R))
\end{aligned}$$

Hence, we have

$$\Gamma^U(\langle u', v' \rangle \mid A) - \Gamma^U(\langle u', v' \rangle \mid A \cup \{\langle u^*, v^* \rangle\}) = \sum_{R \in \mathcal{R}_{v'}} \pi(R) p_{u',v'} p_{u^*,v^*} \mathbb{I}(v^* \in R) = \sum_{R \in \mathcal{R}_{v'} \cap \mathcal{R}_{v^*}} \pi(R) p_{u',v'} p_{u^*,v^*}.$$

The proof is complete. □

Proof of Lemma 5.6. For the lower bound maximization, the worst-case time for identifying the best edge is $O(|S| \min\{n, \sum_{R \in \mathcal{R}} |R|\})$. When updating the marginal gains of each candidate edge after selecting $\langle u^*, v^* \rangle$, we traverse the nodes in every RR set in $\mathcal{R}_{v^*}$, and for each node, we update the marginal gains of the candidate edges targeting at it, which is at most $O(|S|)$. Hence, the overall time complexity of the selection algorithm on $\Gamma^L(\cdot)$ is $O(k\,|S| \sum_{R \in \mathcal{R}} |R|)$. The selection on $\Gamma^U(\cdot)$ is similar, but since we only maintain the marginal gains of nodes but not edges, it only costs $O(k \sum_{R \in \mathcal{R}} |R|)$. □

Lemma A.1 (Adapt from OPIM-C [31]). *Define*

$$\rho_l^L(A) = \left(\left(\sqrt{\Gamma_2^L(A) + \frac{2\zeta}{9}} - \sqrt{\frac{\zeta}{2}} \right)^2 - \frac{\zeta}{18} \right) \cdot \frac{n}{\theta_2},$$

and

$$\rho_u^L(A_L^\circ) = \left(\left(\sqrt{\frac{\Gamma_1^L(A)}{1 - 1/\mathrm{e}} + \frac{\zeta}{2}} + \sqrt{\frac{\zeta}{2}} \right)^2 \right) \cdot \frac{n}{\theta_1}.$$

When $\zeta = \log(9/\delta)$, we have

$$\Pr[\rho_l^L(A) \geq \rho^L(A)] \geq 1 - \frac{\delta}{9}$$
$$\Pr[\rho_u^L(A_L^\circ) \geq \rho^L(A_L^\circ)] \geq 1 - \frac{\delta}{9}.$$

The same results also hold for $\Gamma^U(\cdot)$.

Proof. Let $R_1, R_2, \ldots, R_\theta$ be the sequence of sampled random RR sets and A be an arbitrary set of edges. Define $x_i = (1 - \prod_{u \in S \setminus R}(1 - \mathbf{p}_u w(u, R, A)))\pi_0(R)$, $p = \rho^L(A)/n$ and $M_i = \sum_{j=1}^{i}(x_j - p)$. We will show that the sequence of $M_1, M_2, \ldots, M_\theta$ forms a martingale. First, as $\mathbb{E}[x_i] = p$, we have $\mathbb{E}[|M_i|] = 0 < +\infty$. Second, since the choices of the root v and the sampled realization are independent of the sequence, we have

$$\begin{aligned}\mathbb{E}[M_i | M_1, M_2, \ldots, M_{i-1}] &= \mathbb{E}[M_{i-1} + (x_i - p) | M_1, M_2, \ldots, M_{i-1}] \\ &= M_{i-1} + \mathbb{E}[x_i | M_1, M_2, \ldots, M_{i-1}] - p \\ &= M_{i-1} + \mathbb{E}[x_i] - p = M_{i-1}.\end{aligned}$$

Hence, the sequence of $M_1, M_2, \ldots, M_\theta$ forms a martingale. Then following the same derivation of OPIM-C [31], the lemma holds. □

Proof of Theorem 5.8. The approximation guarantee of the sandwich strategy holds only when the following three events are true:

(1) $\sigma^L(A_L, \mathbf{p}) \geq (1 - 1/\mathrm{e} - \varepsilon)\sigma^L(A_L^\circ, \mathbf{p})$
(2) $\sigma^U(A_U, \mathbf{p}) \geq (1 - 1/\mathrm{e} - \varepsilon)\sigma^U(A_U^\circ, \mathbf{p})$
(3) $(1 - \gamma)\sigma(A, \mathbf{p}) \leq \hat{\sigma}(A, \mathbf{p}) \leq (1 + \gamma)\sigma(A, \mathbf{p})$ for all A in $\{A_L, A_H, A_U\}$.

Combining Lemma A.1 and Lemma 5.7, $\rho^L(A_L) \geq \rho^L(A_L^\circ)$ holds with probability $1 - \frac{\delta}{3}$. Then

$$\begin{aligned}\sigma^L(A_L, \mathbf{p}) - \sigma(\mathbf{p}) \geq& (1 - 1/\mathrm{e} - \varepsilon)(\sigma^L(A_L^\circ, \mathbf{p}) - \sigma(\mathbf{p})) \\ \sigma^L(A_L, \mathbf{p}) \geq& (1 - 1/\mathrm{e} - \varepsilon)\sigma^L(A_L^\circ, \mathbf{p}) + (1/\mathrm{e} + \varepsilon)\sigma(\mathbf{p}) \geq (1 - 1/\mathrm{e} - \varepsilon)\sigma^L(A_L^\circ, \mathbf{p}).\end{aligned}$$

Following the same derivation, $\sigma^U(A_U, \mathbf{p}) \geq (1 - 1/\mathrm{e} - \varepsilon)\sigma^U(A_U^\circ, \mathbf{p})$ also holds with probability at least $1 - \frac{\delta}{3}$. For the final estimation stage, since we ensure a γ-multiplicative error with probability $1 - \frac{\delta}{9}$ and there are three sets to evaluate. Condition (3) holds with probability at least $1 - \frac{\delta}{3}$. Finally, the approximation guarantee holds with probability at least $1 - \delta$ by the union bound. □

Proof of Theorem 5.9. We mainly decompose the algorithm into two stages: (1) bounding functions maximization and (2) estimation. Stage (1) can be further divided into the sampling phase and selection phase. For sampling, by the results of OPIM-C [31], an expected number of $O(\frac{(k \ln n + \ln(1/\delta))n}{\varepsilon^2 \mathrm{OPT}_L})$ RR sets are sampled. The time cost of selection phase is dominated by the lower bound maximization, so the running time is $O(k|S| \sum_{R \in \mathcal{R}} |R|)$. For stage (2), by utilizing the generalized stopping rules [44], an expected number of $O(\frac{n \ln(1/\delta)}{\gamma^2 \sigma(A^*, \mathbf{p})})$ random RR sets will be generated. Putting it together yields the final result. □