Robust Fair Influence Maximization under Multiple Community Partitions

TIANYOU GAO, Kyoto University, Japan
TAKAYUKI ITO, Kyoto University, Japan

Fair Influence Maximization (FIM) is an important extension of the Influence Maximization (IM) problem that incorporates community-level fairness constraints. Most existing FIM formulations assume a given community partition P and a fairness parameter $\boldsymbol{\alpha}$, which governs the trade-off between influence spread and fairness. However, real-world social networks seldom admit a unique, ideal partition that captures community structures accurately. To handle the diversity of community partitions and fairness parameter settings in FIM, we propose the Robust Fair Influence Maximization (RFIM) problem. Given a set of partitions $\mathcal{P}$ and corresponding fairness parameters, RFIM aims to maximize the worst-case ratio between the achieved fair influence objective and the optimal value attainable under each individual partition. We show that RFIM is hard to approximate within any constant factor, even when the seed budget is allowed to exceed the original limit by up to a logarithmic threshold. Fortunately, further surpassing this threshold enables a $(1-1/e, \ln|\mathcal{P}|+O(1))$ bicriteria approximation via the Submodular Saturation framework proposed by Krause et al. Combining this framework with the Hit-and-Stop (HIST) algorithm, we propose Saturate-HIST (S-HIST), a tailored algorithm for RFIM that achieves a $(1-1/e-\varepsilon/OPT)$-approximation with high probability in a bicriteria sense. To improve the efficiency of S-HIST, we further develop a subroutine algorithm, Saturated Generalized HIST (SG-HIST), tailored for its core procedure of maximizing a truncated fair influence objective under a saturated seed budget. We extensively evaluate the performance and scalability of S-HIST on both real-world and synthetic datasets. The experimental results demonstrate that our algorithm consistently achieves robust and competitive performance across diverse datasets and community partitions compared to baseline methods.

CCS Concepts: • **Information systems** → **Social networks**; *Information retrieval diversity*; • **Theory of computation** → *Approximation algorithms analysis*; • **Mathematics of computing** → Submodular optimization and polymatroids.

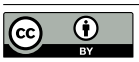
Additional Key Words and Phrases: Fair Influence Maximization, Social Networks, Approximation Algorithm, Robust Optimization

ACM Reference Format:
Tianyou Gao and Takayuki Ito. 2026. Robust Fair Influence Maximization under Multiple Community Partitions. *Proc. ACM Manag. Data* 4, 1 (SIGMOD), Article 78 (February 2026), 25 pages. https://doi.org/10.1145/3786692

1 Introduction

From political campaigns to commercial marketing, a central challenge is to strategically select a small set of initial influencers to maximize influence spread through word-of-mouth diffusion in online social networks (OSNs). This problem, known as Influence Maximization (IM), has been extensively studied over the past two decades [7, 10, 12, 15, 22, 23, 27, 33, 34, 38, 49–51, 56]. Given a graph G, a cardinality constraint k, and an Influence Diffusion Model (IDM), typically the Independent Cascade (IC) [10] or Linear Threshold (LT) [12] model, the objective of IM is to select a set of k seed nodes that maximizes the expected number of influenced individuals after the

Authors' Contact Information: Tianyou Gao, Kyoto University, Kyoto, Japan, gao.tianyou.66z@st.kyoto-u.ac.jp; Takayuki Ito, Kyoto University, Kyoto, Japan, ito@i.kyoto-u.ac.jp.



ACM 2836-6573/2025/2-ART78
https://doi.org/10.1145/3786692

propagation process [27]. Not only has the classical IM problem attracted substantial attention, but numerous variants have also been actively studied, such as competitive IM [5, 35], dynamic IM [39, 52], online IM [30, 32, 49], and augmented IM [13, 14, 16].

However, many existing approaches fail to account for the diverse characteristics of different community groups, which play a critical role in achieving fairness in applications such as health interventions and suicide/HIV prevention [41]. The lack of fairness in IM can result in racial minority or LGBTQ+ communities being unfairly allocated a disproportionately small share of influence [19, 54]. To address this, researchers have proposed various formulations of Fair Influence Maximization (FIM) [1, 2, 18, 43, 44, 47, 54, 57]. Most FIM formulations assume the existence of a predefined community partition P in OSNs. In addition, some of them introduce parameters $\boldsymbol{\alpha}$ that balance influence spread and fairness [21, 41, 45]. FIM maximizes the objective $\sigma(S, P, \boldsymbol{\alpha})$ over seed sets S with $|S| \leq k$, given P and $\boldsymbol{\alpha}$. See Section 3.2.1 for formal definitions. However, in real-world scenarios, multiple plausible community partitions may exist based on attributes such as gender, race, age, or education, among others [20]. The appropriate fairness parameters may also vary across these partitions, e.g., one might prioritize fairness in some partitions and prioritize influence in others [21]. Moreover, due to privacy concerns (e.g., users may choose to remain anonymous in OSNs) and the inherent ambiguity of community detection (e.g., different methods may produce vastly different partitions), obtaining a unique and accurate partition is often infeasible even when the complete network structure is available [20]. These challenges reveal a key limitation of existing FIM definitions, i.e., their reliance on fixed and idealized community partitions and fairness parameter settings, which may not reflect the inherent ambiguity and variability observed in real-world networks.

Building on the FIM setup above, we now introduce the **Robust Fair Influence Maximization (RFIM)** problem. To the best of our knowledge, this is the first study that integrates both robustness and fairness in the context of IM. In the RFIM problem, given a social network graph G and a cardinality constraint k, we consider a collection of l non-overlapping community partitions $\mathcal{P} = \{P_1, \cdots, P_l\}$. For each partition P_q ($q \in [l]$), there exists a corresponding fairness parameter $\boldsymbol{\alpha}_q$. For each $q \in [l]$, the pair $(P_q, \boldsymbol{\alpha}_q)$ induces a FIM instance; we call this a subproblem of RFIM and denote its objective by $\sigma_q(\cdot)$ and its k-optimal seed set by S_q°. Without loss of generality, we consider two partitions distinct if they are associated with different fairness parameters, even when their community structures are identical.

We now formalize RFIM as a robust objective over all subproblems. In contrast to existing FIM definitions, we define the RFIM objective as the worst-case ratio between the fair influence objective value for the selected seeds in each partition and the optimal value for that partition, i.e., $\rho(S) = \min_{q \in [l]} \frac{\sigma_q(S, P_q, \boldsymbol{\alpha}_q)}{\sigma_q(S_q^\circ, P_q, \boldsymbol{\alpha}_q)}$, where S_q° is the optimal seed set of the subproblem σ_q. By strategically selecting at most k seeds, we aim to maximize the objective, i.e., $\max_{|S| \leq k} \rho(S)$. Our definition follows the general approach in operations research for describing algorithmic robustness by [4], and can be seen as an extension of the Robust Influence Maximization (RIM) in [9, 24, 25] in the context of fairness-aware settings. A straightforward alternative is to aggregate multiple partitions by assigning weights to per-partition objectives and maximizing the resulting weighted sum. This idea leads to the Composite Community-aware Diversified Influence Maximization (CC-DIM) formulation proposed by Guo et al. [21]. However, this formulation requires pre-specified weights and optimizes a weighted average across partitions rather than providing worst-case robustness, which is our focus.

Example 1.1 (RFIM). Fig. 1 illustrates how different community partitions in OSNs can lead to divergent fairness outcomes. The upper-left graph shows the original network topology under the IC model, with edge probabilities annotated on its edges. The remaining graphs show three

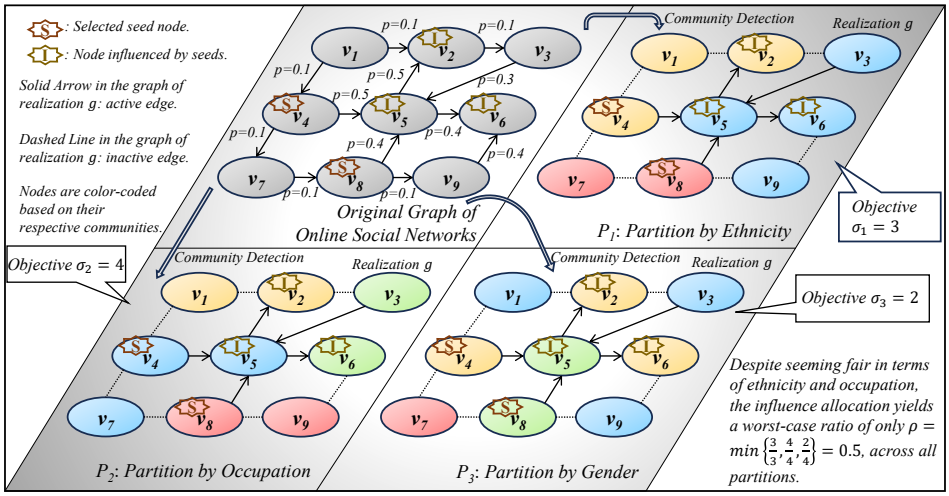


Fig. 1. Example of multiple community partitions and their impact on fair influence.

illustrative partitions (ethnicity, occupation, gender) for visual clarity. Nodes are color-coded to indicate community membership under each attribute. Nodes marked S and I denote the selected seeds and the influenced (non-seed) nodes, respectively. For simplicity, only one deterministic realization is shown, in which influence propagates exclusively along the solid edges displayed in the partition-specific graphs. We adopt a simple fairness objective: a community receives a unit score if at least one of its nodes is influenced. While influence appears fair under the ethnicity- and occupation-based partitions (achieving maximum scores of 3 and 4, respectively), it fails to reach certain communities under the gender-based partition. As shown in the lower-right graph, the communities with red- and blue-coded nodes (different gender groups) receive no influence, resulting in an objective value of only 2 out of a maximum of 4. This motivates adopting a robust objective across plausible partitions.

Remark 1 (Generality and practicality of partitions). This remark clarifies the scope of input partitions. Our framework does not assume that input partitions are single-attribute or unique. In practice, partitions may be induced by multiple attributes jointly (e.g., socioeconomic status combining income, education, and occupation) or by different community detection methods and parameter choices, which often yield complementary views of fairness. We retain multiple plausible partitions rather than collapsing them into a single aggregate to avoid committing to prior weights that may mask minority groups, and RFIM explicitly pursues robustness across such plausible partitions.

Let n be the number of nodes in G. Following the existing FIM definitions [21, 41, 45, 57], we assume that each objective $\sigma_q(S, P_q, \boldsymbol{\alpha}_q)$ is monotone and submodular with respect to S. Although our definition presents RFIM with non-overlapping partitions for exposition, P_q may include overlapping communities, and as long as each objective $\sigma_q(S, P_q, \boldsymbol{\alpha}_q)$ is monotone and submodular in S, all subsequent analysis in this paper extends directly to overlapping community settings. To facilitate our analysis, we adopt two formulations of fairness in this paper: the simple linear aggregation formulation, and the welfare-based formulation. Our analysis readily extends to other existing submodular fairness formulations. However, as in all robust optimization problems, the worst-case ratio objective $\rho(S)$ is not submodular. Even worse, under the "mildest" conditions,

where the objective functions σ_q for all $q \in [l]$ are defined as in [41] with a unified inequality aversion parameter $\alpha \in (0, 1)$, we prove that for any small positive constant $\varepsilon < 0.25$, it is NP-hard to obtain an $O(n^{0.25-\varepsilon})$-approximation by selecting at most $(1 + \ln l)k$ nodes, exceeding the given constraint k by a logarithmic factor of $l = |\mathcal{P}|$. However, once we further allow selecting a constant-factor larger extra seed set, i.e., a factor of $\ln l + O(1)$, a much better bicriteria approximation can be achieved. Specifically, a $(1 - 1/e)$-approximation can be obtained by the Saturate Greedy framework [24, 29], which compares the objective value of the selected seeds against the optimum achievable with k seeds.

To adapt the Saturate Greedy framework to our RFIM problem, we propose the **Saturate-HIST (S-HIST)** algorithm, which uses binary search to maximize a surrogate submodular objective that serves as its core subroutine. However, directly solving the surrogate objective using hill-climbing heuristics with Monte Carlo (MC) simulations, as in [24], is computationally inefficient. To address this, we leverage the idea of Hit-and-Stop (HIST), which builds upon the efficient Reverse Influence Sampling (RIS) technique [7]. We propose a subroutine algorithm, **Saturated Generalized HIST (SG-HIST)**, which solves a saturated fixed-budget variant of the IM problem, contrasting with the seed minimization approach in [24]. Since the surrogate objective function is monotone and submodular, our SG-HIST ensures that the main algorithm S-HIST remains efficient and can guarantee a bicriteria approximation ratio of $(1 - 1/e - \varepsilon/OPT, \ln l + O(1))$ with high probability. Our SG-HIST algorithm is not a straightforward adaptation of the original HIST, as it operates under a saturated seed budget and optimizes a truncated objective that cannot be estimated without bias. Accordingly, we modify key components of HIST to accommodate these changes.

We evaluate S-HIST against several baselines on six real-world and eleven synthetic datasets spanning a wide range of sizes to assess effectiveness, efficiency, and scalability. Across all real-world datasets, S-HIST consistently outperforms or matches the baselines and shows higher stability under variations in data and parameter settings. On synthetic datasets, performance differences are small across methods, echoing observations in prior work [24] and highlighting a practical question: when is Saturate Greedy warranted, and when might a simpler approach (e.g., a basic greedy) suffice for maximizing the worst-case ratio objective?

Our contributions are summarized as follows:

(1) We define the RFIM problem, which considers multiple community partitions to model real-world uncertainty in the FIM problem.
(2) We establish NP-hardness results for approximating the RFIM problem, showing that it admits a poor approximation ratio even under the relaxed constraint of a saturated seed set of size $(1 + \ln l)k$.
(3) We propose S-HIST, which applies the Saturate Greedy framework to RFIM and optimizes a robust submodular surrogate via a HIST-based subroutine, SG-HIST.
(4) For SG-HIST, we redesign HIST for a saturated fixed-budget objective and a truncated fairness objective that lacks an unbiased RIS estimator, introducing modified sampling/stopping rules and objective-evaluation procedures.
(5) We prove that S-HIST achieves a $(1 - 1/e - \varepsilon/OPT)$-approximation with probability at least $(1 - \delta)$ when allowed a $(\ln l + O(1))$-factor enlargement of the seed budget.
(6) We conduct extensive experiments on six real-world and eleven synthetic datasets, demonstrating the effectiveness, efficiency, scalability, and robustness of S-HIST compared with baseline strategies.

Organization: In Section 2, we review related work. In Section 3, we formulate the RFIM problem. In Section 4, we present our algorithm S-HIST and its subroutine algorithm SG-HIST. In Section 5,

we present our experimental settings and results. In Section 6, we conclude our work and discuss future directions.

2 Related Work

In this section, we review related work on IM, FIM, and RIM.

2.1 Influence Maximization

The IM problem was proposed by Kempe et al. [27], aiming to maximize the expected spread of influence under a given IDM by strategically selecting a set of seed nodes. Since its inception, a large body of work has focused on improving the computational efficiency of IM algorithms [7, 10, 12, 15, 22, 23, 38, 49–51, 56]. A key breakthrough was the RIS technique by Borgs et al. [7], which significantly reduces the computational complexity of IM. Building upon the RIS framework, subsequent studies proposed efficient IM algorithms with provable guarantees, including TIM/TIM+ [51], IMM [50], SSA/D-SSA [38], and OPIM-C [49]. These methods achieved a $(1-1/e-\varepsilon)$-approximation with at least $1-1/n$ probability and a complexity of $O(k(n+m)\log n/\varepsilon^2)$. Guo et al. [22, 23] proposed the HIST algorithm, a state-of-the-art method that addresses the scalability bottleneck in high-influence networks by reducing the average size of generated Reverse Reachable (RR) sets without compromising approximation guarantees. All these works focus on maximizing the influence spread and do not consider fairness across different social communities.

2.2 Fair Influence Maximization

Motivated by concerns over inequitable influence spread, various formulations of FIM have been proposed [2, 18, 43, 47, 54], incorporating different fairness notions such as maximin [42, 54], diversity constraints [54], demographic parity [1, 2, 47], and social welfare aggregation [41]. Among them, the social welfare aggregation formulation introduced by Rahmattalabi et al. [41] has been further explored in several recent studies. This formulation is considered for modeling the trade-off between overall influence and fairness across communities. However, their approach relies on MC simulations, restricting its applicability to small-scale networks. To overcome this limitation, Rui et al. [44, 45] proposed a scalable algorithm that constructs an unbiased estimator of fairness objective via Taylor expansion, leveraging the RIS technique for efficient estimation. Subsequent work by Wang et al. [57] further generalized this approach by proposing the twice-differentiable concave fairness framework. However, all the aforementioned FIM models assume a single predefined community partition and a fixed fairness parameter, which fail to reflect the multiple plausible community structures in real-world networks. Guo et al. [21] proposed the CC-DIM problem, which aggregates fairness objectives across partitions through a weighted summation. However, CC-DIM requires pre-specified weights for each community in each partition and focuses on maximizing a weighted average of fairness objectives, rather than ensuring robustness across all partitions in the worst-case sense, as studied in this paper.

2.3 Robust Influence Maximization

RIM addresses settings with uncertainty in influence probabilities or IDM [9, 24]. Chen et al. [9] proposed the solution-dependent algorithms and sampling strategies to mitigate parameter uncertainty. He et al. [24] developed a more general framework, established strong hardness results, and proposed a bicriteria approximation algorithm. Their optimization techniques serve as a foundation for our study. In addition, Anari et al. [3] studied robust submodular maximization under combinatorial constraints and proposed bicriteria approximation algorithms that achieve near-optimal guarantees in both offline and online settings. Chen et al. [8] proposed a general reduction from robust to stochastic optimization and applied it to RIM by computing randomized

Table 1. Frequently Used Notations

Notation	Description
$G = (V, E)$	Social network graph with n nodes and m edges
k	Cardinality constraint on the number of seed nodes
ε	Error tolerance parameter
δ	Failure probability parameter
$\mathcal{P}$	Set of candidate community partitions $\{P_1, \ldots, P_l\}$
P_q	The q-th community partition $\{C_{q,1}, \ldots, C_{q,j_q}\}$
$C_{q,j}$	The j-th community in partition P_q
$n_{q,j}$	Size of community $C_{q,j}$
$\boldsymbol{\alpha}_q$	Fairness parameter vector for partition P_q
$\alpha_{q,j}$	Fairness parameter for community $C_{q,j}$
$F_{q,j}(\cdot)$	Fairness utility function for community $C_{q,j}$
$\sigma_q(S)$	Expected fair influence under partition P_q
S_q°	Optimal seed set for maximizing $\sigma_q(S)$
$\rho(S)$	Robust fair objective achieved by S over all partitions
S°	Optimal seed set for maximizing $\rho(S)$
$\mathcal{R}$	Collection of sampled G-RR sets $\{R^1, \ldots, R^\theta\}$

approximate solutions. Kalimeris et al. [26] studied RIM under hyperparametric models, where edge probabilities are determined by node features and a low-dimensional global parameter. More recently, Saha et al. [46] extended RIM to dynamic networks by proposing the RIME framework, which combines multiplicative weight updates with greedy selection to maintain high-quality seed sets under hyperparametric uncertainty and network evolution. While these works primarily focus on maximizing influence spread under uncertainty in network topology and edge probabilities, they do not consider fairness or account for the diverse community structures that are central to our RFIM problem.

3 Problem Definition

In this section, we introduce several foundational concepts and formally define the RFIM problem.

3.1 Preliminaries

We present the concepts of IDM, IM and approximate solutions in this subsection. Table 1 summarizes the notation used in this paper.

3.1.1 Influence Diffusion Model. We model a social network as a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each directed edge $e = (u, v) \in E$ is associated with an influence probability $p_e \in (0, 1]$, representing the likelihood that an active node u successfully activates its neighbor v.

Given an initial seed set $S \subseteq V$, the diffusion process unfolds stochastically. In this paper, we specifically consider the Discrete-time Independent Cascade (DIC) model [28], where diffusion proceeds in discrete rounds. Initially, only nodes in S are active. In each round, every newly activated node u independently attempts to activate each of its inactive neighbors v, where the attempt succeeds with probability p_e associated with edge (u, v). Each edge permits only one activation attempt during the diffusion process. We model a realization g as a sampled instance of the network, where for each edge (u, v), we independently flip a biased coin with success probability p_e to decide whether (u, v) is included in g. We denote the set of all possible realizations by $\mathcal{G}$. Thus, the

probability of g sampled from $\mathcal{G}$ is $\Pr[g] = \prod_{e \in E_g} p_e \prod_{e \in E \setminus E_g} (1 - p_e)$, where E_g is the set of edges included in the realization g. Given a realization $g \in \mathcal{G}$, the influence process from a seed set S corresponds to visiting all nodes reachable from S through E_g.

3.1.2 Influence Maximization. Given an IDM, the influence spread $\mathrm{Inf}(S)$ measures the expected number of nodes activated by the seed set S. Formally, it is defined as $\mathrm{Inf}(S) = \mathbb{E}_{g \sim \mathcal{G}}[|\Gamma^g(S)|]$, where $\Gamma^g(S)$ denotes the set of nodes reachable from S in the realization g. The IM problem seeks to select a seed set S of size at most k that maximizes the expected influence spread: $\max_{S \subseteq V, |S| \leq k} \mathrm{Inf}(S)$.

3.1.3 Approximate Solutions. A set function $f : 2^V \to \mathbb{R}$ is called submodular if for all $S \subseteq T \subseteq V$ and $v \in V \setminus T$, it holds that $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$. Submodularity captures the diminishing returns property, where the marginal gain from adding an element to a set decreases as the set grows. Moreover, a set function is monotone if $f(S) \leq f(T)$ whenever $S \subseteq T$.

It is known that the influence spread $\mathrm{Inf}(\cdot)$ under the IC (including DIC) or LT model is non-negative, monotone, and submodular [27]. As a consequence, the hill-climbing greedy algorithm, which iteratively adds the node with the largest marginal gain, achieves a $(1 - 1/e)$-approximation guarantee for maximizing $\mathrm{Inf}(S)$ [36]. Moreover, since exactly computing $\mathrm{Inf}(S)$ is #P-hard [10], implementations rely on approximation via sampling, and the guarantee becomes $(1 - 1/e - \varepsilon)$ with high probability.

3.2 Robust Fair Influence Maximization

Recall the definitions introduced in Section 1. $\mathcal{P} = \{P_1, \ldots, P_l\}$ denotes a set of community partitions on the graph G. Each partition $P_q \in \mathcal{P}$ ($q \in [l]$) consists of $j_q = |P_q|$ communities, i.e., $P_q = \{C_{q,1}, \ldots, C_{q,j_q}\}$. For each partition P_q, we associate a fairness configuration vector $\boldsymbol{\alpha}_q = (\alpha_{q,1}, \ldots, \alpha_{q,j_q})$, where each $\alpha_{q,j}$ is a fairness parameter associated with community $C_{q,j}$.

3.2.1 Fair Influence under a Single Partition. When only a single partition is considered, the problem reduces to the FIM setting. Given a seed set S, the expected fair influence under a single partition P_q is defined as $\sigma_q(S, P_q, \boldsymbol{\alpha}_q) = \sum_{C_{q,j} \in P_q} F_{q,j}(S)$, where $F_{q,j}$ is the expected fair influence in community $C_{q,j}$:

$$F_{q,j}(S) = f_{q,j}\left(\sum_{g \in \mathcal{G}} \Pr[g] \cdot \left|\Gamma^g(S) \cap C_{q,j}\right|\right), \tag{1}$$

and $f_{q,j}(x)$ represents the fair influence in $C_{q,j}$ as a function of the expected number x of activated nodes in $C_{q,j}$. For convenience, we abbreviate $\sigma_q(S, P_q, \boldsymbol{\alpha}_q)$ as $\sigma_q(S)$ in the remainder of this paper, since the community partition P_q and fairness parameters $\boldsymbol{\alpha}_q$ are essentially "tied" to the subproblem σ_q. In this paper, we consider two formulations of $f_{q,j}(\cdot)$: a simpler linear aggregation form $f^L_{q,j}(\cdot)$, and a more general welfare-based form $f^W_{q,j}(\cdot)$.

Linear Aggregation Fairness ([21]). A straightforward approach is to define fairness as the weighted sum of the influence spread across communities. I.e., for each community $C_{q,j}$,

$$f^L_{q,j}(x) = \alpha_{q,j} \cdot x, \tag{2}$$

where $\alpha_{q,j} \in (0, 1)$ is an adjustable weight associated with community $C_{q,j}$. The weight $\alpha_{q,j}$ can be tuned according to the importance of community $C_{q,j}$, for example, assigning larger weights to communities with smaller size to promote balance.

Welfare-Based Fairness ([41]). To capture fairness more explicitly, we adopt a welfare-based formulation based on social welfare theory:

$$f^W_{q,j}(x) = {n_{q,j}}^{1-\alpha_{q,j}} \cdot x^{\alpha_{q,j}}, \tag{3}$$

where $n_{q,j} = |C_{q,j}|$ and $\alpha_{q,j} \in (0, 1)$ controls the level of inequality aversion for each community $C_{q,j}$. When $\alpha_{q,j}$ is close to 1, the formulation prioritizes maximizing total activation; when $\alpha_{q,j}$ approaches 0, it emphasizes equitable activation across communities. For clarity, we define the fair influence function $F^L_{q,j}$ and $F^W_{q,j}$ based on their respective fairness formulations, $f^L_{q,j}$ and $f^W_{q,j}$.

Similar to general IM, under either the linear aggregation formulation (2) or the welfare-based formulation (3), the fair influence function $\sigma_q(S)$ is non-negative, monotone, and submodular with respect to the seed set $S \subseteq V$ [41]. Consequently, by applying the classical greedy algorithm, we can achieve a $(1 - 1/e - \varepsilon)$-approximation for maximizing $\sigma_q(S)$ under a single community partition [36].

3.2.2 Fair Influence under Multiple Partitions. Instead of optimizing fair influence under a single fixed partition, we consider a robust objective that simultaneously accounts for multiple plausible partitions. Given a set of l partitions $\mathcal{P}$, the robust fair influence of a seed set S is defined as the worst-case relative performance across all partitions in $\mathcal{P}$:

$$\rho(S) = \min_{q \in [l]} \left\{ \frac{\sigma_q(S)}{\sigma_q(S^\circ_q)} \right\}, \tag{4}$$

where $\sigma_q(S)$ is the fair influence of S under partition P_q, and $\sigma_q(S^\circ_q)$ is the optimal fair influence achievable for P_q under the cardinality constraint $|S| \leq k$. Note that $\sigma_q(S^\circ_q)$ is a fixed constant for each q. Formally, for each partition P_q, the optimal seed set is $S^\circ_q = \text{argmax}_{|S| \leq k} \sigma_q(S)$. The ultimate goal of the RFIM problem is to select a seed set S° of size at most k that maximizes the robustness objective $\rho(\cdot)$:

$$S^\circ = \underset{|S| \leq k}{\text{argmax}}\, \rho(S). \tag{5}$$

Compared to standard FIM that optimizes fair influence under a single partition, RFIM seeks a seed set that guarantees reasonable fairness across all considered partitions simultaneously. However, when extending to multiple partitions, the robustness objective $\rho(S)$, which measures the minimum relative fair influence across partitions, no longer preserves submodularity even though each individual $\sigma_q(S)$ is monotone and submodular. This is because the pointwise minimum of multiple submodular functions is, in general, not submodular.

Additionally, since the IM problem is NP-hard, the exact computation of $\sigma_q(S^\circ_q)$ is intractable. Following the approach in [24], we obtain a $(1 - 1/e)$-approximate solution S^g_q using the greedy heuristic, and use $\sigma_q(S^g_q)/(1-1/e)$ as a surrogate for $\sigma_q(S^\circ_q)$ (omitting the additional estimation error ε for simplicity). Throughout this paper, we treat $\sigma_q(S^\circ_q)$ as a fixed input parameter for convenience, and retain the notation $\sigma_q(S^\circ_q)$ rather than replacing it with $\sigma_q(S^g_q)/(1 - 1/e)$.

3.2.3 Hardness. To formally establish the computational hardness of RFIM, we analyze the problem under both types of fair influence objectives defined in (2) and (3). For each formulation, we show that RFIM is NP-hard to approximate within any meaningful factor, even when the seed budget constraint is slightly relaxed.

We begin by establishing the hardness result for the linear aggregation formulation via a reduction from the gap version of the set cover problem.

Theorem 3.1 (Hardness for Linear Aggregation Fairness). *Let δ, ε be any small positive constants. Under the setting where each subproblem adopts the linear aggregation fair influence objective $f_{q,j}$ defined in* (2)*, the RFIM problem is NP-hard to approximate within a factor of $O(n^{1-poly(\varepsilon)})$ by selecting a seed set of size at most $(1 - \delta) \ln l \cdot k$.*

Next, we turn to the welfare-based formulation, where fair influence is defined through a concave transformation. We establish hardness results under a more restrictive setting in which all communities share a common parameter $\alpha_{q,j}$ across all partitions.

Theorem 3.2 (Hardness for Welfare-based Fairness). *Let δ, ε be any small positive constants. Even under the mildest setting, where all subproblems share identical fair influence objective functions with the same $\alpha \in (0,1)$ as defined in (3), the RFIM problem is NP-hard to approximate within a factor of $O(n^{0.25-poly(\varepsilon)})$ (more precisely, $O(n^{\frac{1}{4}(1-\varepsilon)^2})$) by selecting seed nodes with a size of $|S| \leq (1-\delta)\ln l \cdot k$.*

The above hardness results indicate that the worst-case structure of RFIM over multiple partitions precludes efficient approximation via standard submodular techniques such as greedy selection. To tackle this, we adopt the submodular saturation framework.

4 Method

This section introduces the saturation framework S-HIST and its seed selection subroutine SG-HIST (as illustrated in Fig. 2), followed by a theoretical analysis of their performance guarantees. Due to space limitations, some detailed explanations and proofs are deferred to the appendix.

4.1 The Saturate-HIST Algorithm

To solve the RFIM problem under a bicriteria approximation framework, we propose the S-HIST algorithm, which builds on the idea of saturation in submodular optimization [29]. Furthermore, it integrates the efficient, fixed-budget RIS-based technique, HIST, to determine whether a given threshold in the binary search is feasible, which would be introduced in Section 4.2 later.

Given a truncated threshold $c \in (0,1]$, we define the truncated robust objective function:

$$H^{(c)}(S) = \sum_{q\in[l]} \min\left\{\frac{\sigma_q(S)}{\sigma_q(S_q^\circ)}, c\right\}, \tag{6}$$

which aggregates the truncated ratio of fair influence of the seed set S against the corresponding optimal value $\sigma_q(S_q^\circ)$ across all subproblems. Then, we can convert our optimizing objective from a non-submodular function $\rho(\cdot)$, as defined in (4), to a submodular function $H^{(c)}(\cdot)$.

We present our main algorithm, S-HIST, in Algorithm 1. The algorithm performs a binary search over the target value $c \in (0,1]$ to guide the selection process, as illustrated in the left part of Fig. 2. The following theorem shows that the S-HIST algorithm yields a seed set S^* with a satisfactory approximation ratio with high probability.

Theorem 4.1. *For any $\varepsilon, \delta > 0$, Algorithm 1 returns a seed set S^* of size βk, satisfying*

$$\rho(S^*) \geq (1-1/e)\rho(S^\circ) - \varepsilon$$

with probability at least $1-\delta$.

The proof of Theorem 4.1 relies on the conclusion of Theorem 4.2, which will be discussed later. Notably, instead of incrementally building a seed set until a threshold of $c(l-\varepsilon/3)$ is met, as in existing work of RIM [24], S-HIST fixes the seed budget to $\beta k = \lceil k \ln \frac{6l}{\varepsilon} \rceil$, and then selects a seed set of size βk using SG-HIST (Algorithm 4) to maximize $H^{(c)}(S)$. The rationale behind this modification is that, in incremental selection, estimating fair influence at each step would require repeated MC simulations or RR set regeneration (i.e., repeatedly generating $\mathcal{R}_2$ for each size of seed nodes in Algorithm 6, line 8). Our modification eliminates this computational overhead.

Some recent works on the seed minimization problem reformulate the standard fixed-budget IM setting as a fixed-threshold seed minimization problem [17, 48]. However, under the same target

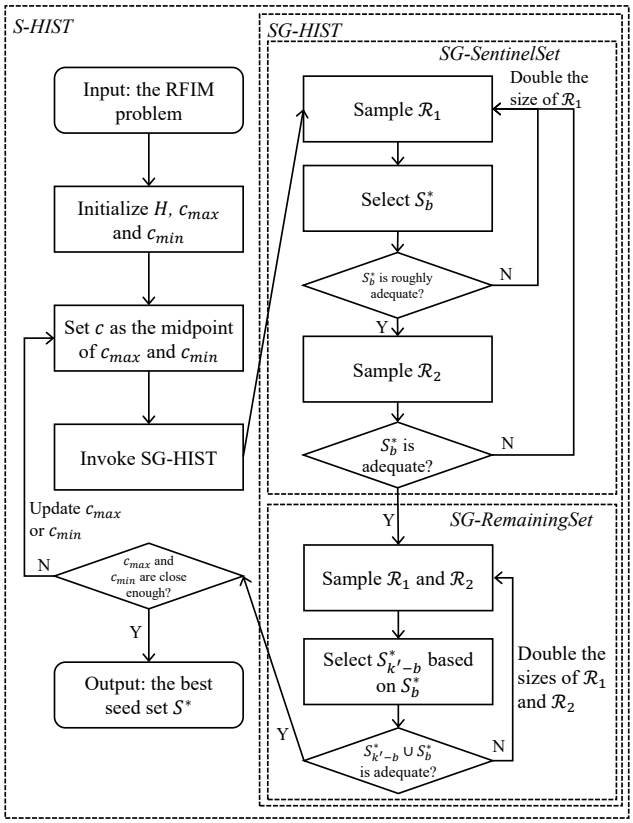


Fig. 2. Algorithmic workflow of RFIM with S-HIST (outer) and SG-HIST (inner). S-HIST converts the non-submodular objective ρ into a saturated, truncated, submodular surrogate $H^{(c)}$ and determines c via binary search. SG-HIST maximizes $H^{(c)}$ in two phases: first selecting a small set of high-influence sentinel seeds, then selecting the remaining seeds. Unlike HIST, SG-HIST uses a truncated objective under a saturated seed budget.

truncation threshold, the theoretically achievable maximum saturation ratio β does not change between these formulations. Therefore, for simplicity, we adopt the fixed-budget formulation as the subroutine problem in this paper.

By comparing Theorem 3.1 (Theorem 3.2) with Theorem 4.1, we observe that, the use of S-HIST improves the approximation guarantee from $O(n^{1-\varepsilon})$ ($O(n^{0.25-\varepsilon})$) to a constant-factor bound of $1-1/e$, by merely increasing the seed budget by an additive factor of $O(1)$ (i.e., from $(1-\delta)\ln l$ to $\ln l + O(1)$).

The approximation ratio achieved by Algorithm 1 is solution-dependent, i.e., $O(1-1/e-\varepsilon/OPT)$, where $OPT = \rho(S^\circ)$. Therefore, to obtain a reasonably good approximation, the input parameter ε should be significantly smaller than OPT. In the following analysis, we assume that $OPT > \varepsilon/(1-1/e)$; otherwise, the error tolerance ε is too large to yield a meaningful approximation and should be reduced. In such cases, even a randomly selected seed set S_{rand} would satisfy $|\rho(S_{rand}) - OPT| \le \varepsilon/(1-1/e)$.

Algorithm 1 Saturation-Hit-and-Stop (S-HIST) Framework

Input: Set of fair influence objectives $\{\sigma_q(S)\}_{q=1}^{l}$, budget k, error tolerance ε, failure probability δ,
Output: Seed set S^*, such that $\rho(S^*) \geq (1-1/e)\rho(S^\circ) - \varepsilon$

$\beta \leftarrow \lceil k \ln \frac{6l}{\varepsilon} \rceil / k$, $\varepsilon_0 \leftarrow \frac{\varepsilon}{6l}$, $\delta_0 \leftarrow \delta / \lceil \log_{5/6} \varepsilon \rceil$
$c_{min} \leftarrow 0$, $c_{max} \leftarrow 1$, $c^* \leftarrow 0$, $S^* \leftarrow \emptyset$
while $c_{max} - c_{min} \geq \varepsilon$ **do**
  $c \leftarrow (c_{min} + c_{max})/2$
  Define $H^{(c)}(S)$ by (6)
  $(S_c, \underline{H}^{(c)}) \leftarrow$ SG-HIST$(H^{(c)}, \beta, k, \varepsilon_0, \delta_0)$ (Algorithm 4)
  if $\underline{H}^{(c)} \geq c(l - \varepsilon/3)$ **then**
    $c_{min} \leftarrow c(1-\varepsilon/3)$, $c^* \leftarrow c$, $S^* \leftarrow S_c$
  else
    $c_{max} \leftarrow c$
  end if
end while
return S^*

4.2 An Efficient Approximation Algorithm

In this subsection, we begin by reviewing the RIS technique and then introduce SG-HIST, a variant of HIST tailored to optimize the saturated objective $H^{(c)}(\cdot)$ studied in this paper (as illustrated in Fig. 2). Since c is fixed throughout this section, we omit the superscript (c) (e.g., in $H^{(c)}$) for brevity when the context is clear.

4.2.1 The Overview of SG-HIST. At a high level, SG-HIST differs from the existing algorithm, HIST, in two key aspects:

(1) It estimates a truncated objective that cannot be estimated without bias.
(2) It adopts a saturated seed set, thereby achieving a factor of $(1 - e^{-\beta})$ instead of the classical $(1 - 1/e)$.

Both differences are crucial for coupling SG-HIST within the submodular saturation framework.

We first introduce an estimator $\hat{H}_{\mathcal{R}}(\cdot)$ for $H(\cdot)$ based on the RIS technique. However, the estimator is biased due to the truncation operator $\min\{\cdot, c\}$ and is therefore not directly applicable to HIST. To address this, SG-HIST employs a tighter precision parameter ε than what would be used under unbiased estimation, thereby requiring more G-RR sets to be sampled. This allows us to control the error tolerance within a target precision level ε_0, even though the estimator is biased. Unlike IMM, which employs a concentration bound under a martingale sequence, we cannot directly apply it to bound the error of H due to its inherent bias. Fortunately, since the estimation of each $\sigma_q(\cdot)$ remains unbiased, concentration inequalities can be applied to derive their bounds. By bounding for the bias introduced by the $\min\{\cdot, c\}$ truncation, we derive valid confidence bounds for the target objective H. In addition, we adapt the early stopping technique used in OPIM-C for H, enabling efficient yet accurate estimation of the approximation ratio. Finally, unlike HIST, which achieves a $(1 - 1/e - \varepsilon)$-approximation, SG-HIST attains a stronger $(1 - e^{-\beta} - \varepsilon)$-approximation owing to its saturated seed selection. This necessitates a modest increase in the number of sampled G-RR sets.

4.2.2 Reverse Influence Sampling for RFIM. RIS [7] is a pivotal technique for IM, significantly reducing the computational overhead compared to MC-simulation-based methods. In our work, the RIS technique is extended to accommodate RFIM objectives. Specifically, we adopt a Generalized-RR (G-RR) set (i.e., a group of RR sets generated from multiple sources under the same realization [21],

Algorithm 2 G-RR-Sampling $(G, \mathcal{P}, S_b^*)$

Initialize the i-th G-RR set R^i as a map indexed by (q, j)
for each partition $P_q \in \mathcal{P}$ **do**
  Sample a realization $g \sim \mathcal{G}$ for P_q
  for each community $C_{q,j} \in P_q$ **do**
    Randomly select a node $v_{q,j}^i \in C_{q,j}$
    Run reverse BFS from $v_{q,j}^i$ in g, truncated if S_b^* is reached
    $R_{q,j}^i \leftarrow$ collected nodes in BFS or $\emptyset$ if truncated by S_b^*
  end for
end for
return R^i

Algorithm 3 MaxCoverage-Greedy $(\mathcal{R}, k)$

Initialize S_0 with the current seed set (e.g., S_b^*) if it is already available; otherwise, set $S_0 \leftarrow \emptyset$
for $a = 1$ to k **do**
  $v \leftarrow \text{argmax}_{u \in V \setminus S_{a-1}} \Lambda_{\mathcal{R}}(u|S_{a-1})$
  $S_a \leftarrow S_{a-1} \cup \{v\}$
end for
return S_k

also known as multi-root RR set [48]) sampling framework, in place of standard RR sets. The procedure is outlined in Algorithm 2, which is self-explanatory. Notably, unlike the approach in [21], we resample an independent realization g for each partition. This modification introduces no additional time complexity, since a BFS from each selected seed is necessary in either case.

Based on G-RR sets sampled by Algorithm 2, we define $\hat{H}_{\mathcal{R}}^{(c)}$ as follow, which is an estimator of the truncated objective $H^{(c)}$:

$$\hat{H}_{\mathcal{R}}^{(c)}(S) = \sum_{q=1}^{l} \min\left\{ \frac{\sum_{C_{q,j} \in P_q} \hat{F}_{q,j}(S, \mathcal{R})}{\sigma_q(S_q^\circ)}, c \right\}, \tag{7}$$

where $\hat{F}_{q,j}(S, \mathcal{R})$ is an estimator of the fair influence of the seed set S in partition q, community j under the generated G-RR set $\mathcal{R}$. The definition of $\hat{F}_{q,j}(S, \mathcal{R})$ depends on the chosen fairness formulation. Here, we present the definition for the linear aggregation case, denoted by $\hat{F}_{q,j}^L(S, \mathcal{R}) = \alpha_{q,j} \cdot n_{q,j} \cdot \frac{1}{\theta} \sum_{i=1}^{\theta} \mathbb{I}(S \cap R_{q,j}^i \neq \emptyset)$, and abbreviate it as $\hat{F}_{q,j}^L(S)$. Similarly, we can define the estimator for the welfare-based case as $\hat{F}_{q,j}^W(S, \mathcal{R})$, which we also abbreviate as $\hat{F}_{q,j}^W(S)$. Due to space limitations, the details are provided in the appendix. For simplicity, we uniformly denote both estimators by $\hat{F}_{q,j}(S)$ in the corresponding context.

4.2.3 Bounding the Surrogate Objective. We first define the generalized coverage as $\Lambda_{\mathcal{R}}^{(q)}(S) = \theta \cdot \sum_{C_{q,j} \in P_q} \hat{F}_{q,j}(S, \mathcal{R})$. Based on this, we introduce lower and upper bounds for the objective σ_q of each subproblem using a similar approach as OPIM-C [49]:

$$\underline{\sigma_q}(S^*) = \left[\left(\sqrt{\Lambda_{\mathcal{R}_2}^{(q)}(S^*) + \frac{2\ln(\frac{1}{\delta_l})}{9}} - \sqrt{\frac{\ln(\frac{1}{\delta_l})}{2}} \right)^2 - \frac{\ln(\frac{1}{\delta_l})}{18} \right] \cdot \frac{1}{\theta_2}, \tag{8}$$

Algorithm 4 SG-HIST ($H^{(c)}$, β, k, ε_0, δ_0)

Input: An objective function $H^{(c)}$, saturation rate β, budget k, error tolerance ε_0, failure probability δ_0

Output: Seed set S^* and the lower bound $\underline{H}^{(c)}(S^*)$, such that $\underline{H}^{(c)}(S^*) \geq (1 - e^{-\beta} - \varepsilon_0)H^{(c)}(S_H^\circ)$

1: Compute H_{min} by (12), then set $\varepsilon_1, \varepsilon_2 \leftarrow \varepsilon_0 \cdot H_{min}/(4l)$, $\varepsilon_3, \varepsilon_4 \leftarrow \varepsilon_0/4$, $\delta_1, \delta_2 \leftarrow \delta_0/4$
2: $S_b^* \leftarrow$ SG-SentinelSet($H^{(c)}, \beta, k, \varepsilon_1, \varepsilon_3, \delta_1$)
3: $(S_d^*, \underline{H}^{(c)}) \leftarrow$ SG-RemainingSet($H^{(c)}, \beta, k, S_b^*, \varepsilon_0, \varepsilon_2, \varepsilon_4, \delta_2$)
4: $S^* \leftarrow S_b^* \cup S_d^*$
5: **return** $S^*, \underline{H}^{(c)}$

$$\overline{\sigma_q}(S_k^\circ) = \left(\sqrt{\overline{\Lambda}_{\mathcal{R}_1}^{(q)}(S_q^\circ) + \frac{\ln(\frac{1}{\delta_u})}{2}} + \sqrt{\frac{\ln(\frac{1}{\delta_u})}{2}} \right)^2 \cdot \frac{1}{\theta_1}, \tag{9}$$

where δ_l, δ_u are the failure probabilities, i.e., the inequality $\sigma_q(S^*) \geq \underline{\sigma_q}(S^*)$ holds with probability at least $1 - \delta_l$; and similarly, the inequality $\sigma_q(S_k^\circ) \leq \overline{\sigma_q}(S_k^\circ)$ holds with probability at least $1 - \delta_u$. In (9), the term $\overline{\Lambda}_{\mathcal{R}_1}^{(q)}(S_q^\circ)$ is the upper bound on the coverage of S_q° on $\mathcal{R}_1$, defined as $\overline{\Lambda}_{\mathcal{R}_1}^{(q)}(S_q^\circ) = \min_{0 \leq a \leq \beta k} \left\{ \Lambda_{\mathcal{R}_1}^{(q)}(S_a^*) + \sum_{v \in maxMC_{\mathcal{R}_1}^{(q)}(S_a^*, k)} \Lambda_{\mathcal{R}_1}^{(q)}(v | S_a^*) \right\}$, where S_a^* denotes the nodes selected during the first a iterations of Algorithm 3 and $maxMC_{\mathcal{R}_1}^{(q)}(S_a^*, k)$ is the set of nodes with the top-k largest marginal coverage on $\mathcal{R}_1$ with respect to S_a^*. By applying this, we can evaluate $\overline{\Lambda}_{\mathcal{R}_1}^{(q)}(S_q^\circ)$ without constructing S_q° explicitly. We further define lower and upper bounds of the truncated objective H as follows:

$$\underline{H}^{(c)}(S^*) = \sum_{q=1}^{l} \min \left\{ \frac{\underline{\sigma_q}(S^*)}{\sigma_q(S_q^\circ)}, c \right\}, \tag{10}$$

$$\overline{H}^{(c)}(S_k^\circ) = \sum_{q=1}^{l} \min \left\{ \frac{\overline{\sigma_q}(S_k^\circ)}{\sigma_q(S_q^\circ)}, c \right\}. \tag{11}$$

4.2.4 The Procedure of SG-HIST. SG-HIST (Algorithm 4) follows the two-phase HIST framework. The first phase (Algorithm 5) selects a small set of high-influence sentinel seed nodes while using only a few G-RR sets. The second phase (Algorithm 6) selects the remaining seeds using more G-RR sets overall, but with smaller average size, since many generated sets hit the sentinel seeds and are discarded without being stored.

Both phases are based on OPIM-C. Accordingly, we outline the core OPIM-C ideas and defer technical details to the appendix due to space constraints. In each phase, we maintain two independent collections of G-RR sets, denoted by $\mathcal{R}_1$ and $\mathcal{R}_2$. Specifically, $\mathcal{R}_1$ is employed to select seed nodes and estimate an upper bound on the optimal value achievable with k seeds, whereas $\mathcal{R}_2$ is used to compute a lower bound for the influence of the selected seed set. If the ratio between the lower and upper bounds exceeds the target approximation ratio threshold, the algorithm terminates and returns the current seed set. Otherwise, the sizes of both $\mathcal{R}_1$ and $\mathcal{R}_2$ are doubled, and the algorithm proceeds to the next iteration.

4.2.5 Bounding the Number of G-RR Sets. An important concern is that the ratio of the lower to the upper bound in (10) and (11) may never reach the target threshold during the iterations. In that case, the algorithm could keep doubling the sizes of $\mathcal{R}_1$ and $\mathcal{R}_2$ without terminating. To guarantee

Algorithm 5 SG-SentinelSet ($H^{(c)}$, β, k, ε_1, ε_3, δ_1)

Set $\theta_0 = 3 \cdot \ln(1/\delta_1)$ and θ_{max} according to (13)
Generate random G-RR sets $\mathcal{R}_1$ with $|\mathcal{R}_1| = \theta_0$
$i_{max} \leftarrow \lceil \log_2(\theta_{max}/\theta_0) \rceil$
for $i = 1$ to i_{max} **do**
  $S_k^* \leftarrow$ MaxCoverage-Greedy($\mathcal{R}_1, k$)
  Compute the rough lower bound $\underline{\hat{H}}(S_a^*)$ by (10) on $\mathcal{R}_1$ and S_k^*, where $1 \le a \le k$
  Compute $\overline{H}(S_k^\circ)$ by (11) on $\mathcal{R}_1$, $\delta_u = \delta_1/(3i_{max})$
  Let b be the maximum number such that $\underline{\hat{H}}(S_a^*)/\overline{H}(S_k^\circ) \ge (1 - (1-1/k)^a - \varepsilon_3)$
  Generate a collection of G-RR sets $\mathcal{R}_2$ with $|\mathcal{R}_2| = |\mathcal{R}_1|$ by invoking Algorithm 2 based on S_b^*
  Compute $\underline{H}(S_b^*)$ by (10) on $\mathcal{R}_2$, $\delta_l = \delta_1/(6i_{max})$
  if $\underline{H}(S_b^*)/\overline{H}(S_k^\circ) \ge (1 - (1-1/k)^b - \varepsilon_3)$ **then**
    return S_b^*
  end if
  Increase the size of $\mathcal{R}_2$ to $4|\mathcal{R}_1|$ and recompute $\underline{H}(S_b^*)$ by (10) on $\mathcal{R}_2$
  if $\underline{H}(S_b^*)/\overline{H}(S_k^\circ) \ge (1 - (1-1/k)^b - \varepsilon_3)$ **then**
    return S_b^*
  end if
  Double the size of $\mathcal{R}_1$
end for

Algorithm 6 SG-RemainingSet ($H^{(c)}$, β, k, S_b^*, ε_0, ε_2, ε_4, δ_2)

Set $\theta_0 = 3 \cdot \ln(1/\delta_2)$ and θ_{max} according to (14)
Generate two G-RR sets $\mathcal{R}_1$ and $\mathcal{R}_2$ of size θ_0, based on S_b^*
$i_{max} \leftarrow \lceil \log_2(\theta_{max}/\theta_0) \rceil$, $k' \leftarrow \beta k$
for $i = 1$ to i_{max} **do**
  Select a size-$(k'-b)$ seed set $S_{k'-b}^*$ from $V \setminus S_b^*$ on $\mathcal{R}_1$ by Algorithm 3, initialized with $S_0 = S_b^*$
  $S_{k'}^* \leftarrow S_b^* \cup S_{k'-b}^*$
  Compute $\overline{H}(S_k^\circ)$ using (11) on $\mathcal{R}_1$ with $\delta_u = \delta_2/(3i_{max})$
  Compute $\underline{H}(S_{k'}^*)$ using (10) on $\mathcal{R}_2$ with $\delta_l = \delta_2/(3i_{max})$
  if $\underline{H}(S_{k'}^*)/\overline{H}(S_k^\circ) \ge (1 - e^{-\beta} - \varepsilon_0)$ **then**
    return $S_{k'-b}^*, \underline{H}(S_{k'}^*)$
  end if
  Double the size of $\mathcal{R}_1$ and $\mathcal{R}_2$
end for
Re-estimate $H(S_{k'}^*)$ via Algorithm 2 in [37] to obtain a (ε_4,δ_2)-approximation $\hat{H}(S_{k'}^*)$, and set the lower bound $\underline{H}(S_{k'}^*)$ to $\hat{H}(S_{k'}^*)/(1+\varepsilon_4)$
return $S_{k'-b}^*, \underline{H}(S_{k'}^*)$

termination, we cap the size of $\mathcal{R}_1$ at $\theta_{\max}$ in both Algorithm 5 and Algorithm 6. We first introduce the following quantities. Let $\sigma_{max} = \max_{q \in [l]} \sigma_q(S_q^\circ)$. Let S_{min} be any size-k set that maximizes $H'(S')$, where $H'(\cdot)$ is the truncated objective under the degenerate realization in which no edges

activate. Set $\sigma_{min} = \max_{q\in[l]} \left\{\sigma_q(S_{min}),\ \sigma_q(S_q^\circ)\,\varepsilon/(1-1/e)\right\}$ and

$$H_{min} = \max\{H(S_{min}),\ \varepsilon\, l/(1-1/e)\}\,. \tag{12}$$

Then, the number of G-RR sets in Algorithm 5 is upper bounded by

$$\theta_{max} = \frac{2\cdot\left[\sqrt{2\ln\frac{6l}{\delta_1}} + \sqrt{\left(\ln\binom{n}{k} + \ln\frac{6l}{\delta_1}\right)\left(2+\frac{2}{3}\varepsilon_1\right)\frac{\sigma_{max}}{\sigma_{min}}}\right]^2}{\varepsilon_1^2\cdot\sigma_{min}}. \tag{13}$$

Intuitively, our θ_{max} in (13) differs from the expression in Equation (3) of the HIST method in [22], as ours includes an additional $\frac{\sigma_{max}}{\sigma_{min}}$ term inside the second square root. This difference arises because our final goal is to optimize $H(\cdot)$ rather than $\sigma_q(\cdot)$. Specifically, when applying the union bound to ensure that the concentration inequality holds for all possible size-b seed sets, we must compare the objective value $\sigma_q(\cdot)$ against $\sigma_q(S_H^\circ)$, rather than $\sigma_q(S_q^\circ)$, as is done in general IM algorithms such as IMM.

Similarly, an upper bound on the number of G-RR sets required by Algorithm 6 is

$$\theta_{max} = \frac{2\sigma_{max}\left[\sqrt{\ln\frac{9l}{\delta_2}} + \sqrt{\left(2\beta+\frac{2}{3}\varepsilon_2\right)\left(\ln\binom{n-b}{n-\beta k} + \ln\frac{9l}{\delta_2}\right)}\right]^2}{\varepsilon_2^2\cdot\sigma_{min}^2}. \tag{14}$$

Compared to the expression in (13), θ_{max} defined by (14) further introduces a new factor, β, in the second square root. However, this also originates from the same reason: our final objective differs from that of standard IM algorithms. As a result, we use a conservative upper bound based on submodularity for the size-βk seed set when comparing its objective value against the size-k seed set S_q°. Full proofs by induction for (13) and (14) are provided in the appendix.

4.2.6 Theoretical Results of SG-HIST. In contrast to HIST, Algorithm 6 requires not only that the final solution S^* achieves the target approximation ratio with respect to $H^{(c)}(\cdot)$, but also that a valid lower bound $\underline{H}^{(c)}(S^*)$ for this objective is certified to meet the target approximation ratio. This requires us to strengthen the theoretical guarantee for the worst-case scenario, where S^* is returned in the final iteration without satisfying the target guarantee via optimistic approximation. To address this, we simply return S^* at termination if no round satisfies the optimistic condition, and perform an additional estimation to compute a lower bound of $H^{(c)}(S^*)$ within an additive error of ε_4 (Algorithm 6, line 14). This additional estimation introduces a new error. Therefore, we adopt a stricter error budget, splitting the total allowed error and failure probability across four components instead of two. Specifically, the values of ε and δ in SG-HIST are divided by 4 (Algorithm 4, line 1) rather than 2, as done in HIST. By applying the union bound over all failure probabilities of possible error sources, we ensure that the final solution S^*, along with its certified lower bound, satisfies the desired approximation guarantee with probability at least $1-\delta_0$. The following theorem states the approximation guarantee achieved by Algorithm 4:

Theorem 4.2. *Given any $\varepsilon_0 > 0$, with probability at least $1-\delta_0$, Algorithm 4 returns a seed set S^* such that*

$$H^{(c)}(S^*) \geq \left(1 - e^{-\beta} - \varepsilon_0/2\right)\cdot H^{(c)}(S^\circ_{H^{(c)}}),$$

where $H^{(c)}(\cdot)$ is defined in (6)*, and $S^\circ_{H^{(c)}}$ denotes the optimal seed set of size k maximizing $H^{(c)}(\cdot)$. Furthermore, the algorithm returns a certified lower bound $\underline{H}^{(c)}$ such that*

$$\underline{H}^{(c)}(S^*) \geq \left(1 - e^{-\beta} - \varepsilon_0\right)\cdot H^{(c)}(S^\circ_{H^{(c)}}).$$

Table 2. Description of Datasets

Dataset	$\|V\|$	$\|E\|$	$\|\mathcal{P}\|$	Labeled
NetHEPT	15.2K	62.8K	4	No
Amazon	335K	1.85M	3	Yes
DBLP	317K	2.10M	4	Yes
Youtube	1.13M	5.98M	3	Yes
Wiki	1.79M	28.5M	3	Yes
LiveJournal	4.00M	69.4M	3	Yes

Theorem 4.2 provides the final component required to establish the theoretical guarantee of our main theorem (Theorem 4.1).

4.3 Complexity

By combining the time complexity of the binary search procedure in Algorithm 1 with that of Algorithm 4, whose analysis follows a similar approach to that in [21, 23], we derive the worst time complexity of S-HIST as

$$O\left(\log(1/\varepsilon)\frac{(\sum_{q\in[l]} j_q)mn\sigma_{max}\cdot\beta(k\log n+\log(1/\delta_0))}{(H_{min}\cdot\sigma_{min}\cdot\varepsilon_0)^2}\right).$$

In the simplified case where l, j_q, and δ_0 are constants, and ε_0 is set to $\varepsilon/\log(1/\varepsilon)$, under the lower-bound assumption on $OPT > \frac{\varepsilon}{1-1/e}$, which further lower bounds H_{min} and $\frac{\sigma_{min}}{\sigma_{max}}$ by $\Omega(\varepsilon)$, the overall worst-case complexity simplifies to $O\left(\frac{\log^4(1/\varepsilon)\cdot mnk\log n}{\sigma_{max}\cdot\varepsilon^6}\right)$. Despite the ε^6 term in the denominator, the algorithm typically runs within an acceptable time frame, owing to the early termination strategy adapted from OPIM-C. That said, the actual runtime still largely depends on the specific characteristics of each instance, as demonstrated in our experiments.

5 Experiment

Our experiments are designed to answer three main questions:

(1) How effective is S-HIST compared with the baseline algorithms?
(2) How efficient is it in practice?
(3) How well does it scale with respect to graph size and the number of partitions?

5.1 Experimental Settings

All experiments are conducted on a Linux server with an AMD EPYC 7502P 32-Core Processor and 251 GB of RAM. To reduce the effect of randomness, each experiment is repeated 10 times, and the average performance is reported over 10,000 sampled diffusion realizations per run. All programs are implemented in C++17 and compiled with the `-O3` optimization. Our implementation is available at https://github.com/TianyouG/RFIM.

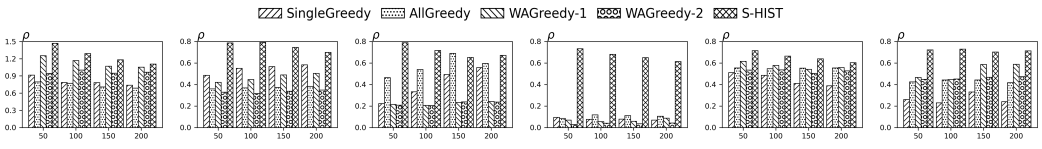
5.1.1 Datasets. We used six real-world datasets of varying sizes from diverse domains, i.e., *NetHEPT*, *Amazon*, *DBLP*, *Youtube*, *Wiki*, *LiveJournal*. *NetHEPT* is available in [11]. The rest of the datasets are from SNAP [31]. We converted all undirected graphs into directed graphs by replacing each undirected edge with two directed edges of opposite directions. Table 2 summarizes the details of the datasets. For additional information, please refer to the appendix.

5.1.2 Community Partitioning. Since this work aims to address the RFIM problem under multiple plausible community partitions, we use not only the ground-truth communities but also apply

various community detection methods to preprocess the data and obtain diverse community partitions. Specifically, we use the following four methods to obtain community partitions:

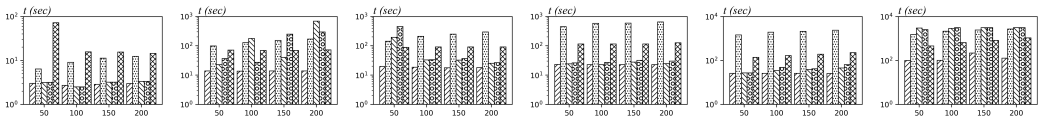
- *Ground-truth:* All datasets except *NetHEPT* come with ground-truth community information. We filter out communities with fewer than 10 nodes to improve the quality of the partition. Some communities in our datasets overlap. Although our formulation can be extended to overlapping communities, for consistency with the non-overlapping setting used in this paper, we adopt a simple voting-based strategy that yields disjoint communities: (1) For nodes with multiple community labels, we assign them the label among their own community labels that appears most frequently among their neighbors. (2) For nodes without any community label, we assign them the label of the community that appears most frequently among their neighbors.
- *Leiden ([53]):* An improved version of the well-known Louvain algorithm [6], designed to produce higher-quality partitions and guarantee well-connected communities based on modularity. We apply *Leiden* to all datasets.
- *Label Propagation ([40]):* A general algorithm for community detection on large graphs. We apply this method to all datasets. For datasets with ground-truth communities (i.e., all datasets except *NetHEPT*), we initialize the community labels using the ground-truth ones after preprocessing with the aforementioned voting-based strategy. For *NetHEPT*, we initialize the labels in two different ways: (1) each node is assigned a unique label; (2) each node is assigned the label obtained from the result of the Leiden algorithm. Although this algorithm is simple and scalable, its clustering results are generally random and may not converge deterministically. In our experiments, we let the algorithm run until convergence to ensure a stable partition.
- *Spectral Clustering ([55]):* A general-purpose community detection method based on the eigenstructure of graph Laplacian matrices, which can be applied to a wide range of networks. However, as this algorithm is computationally expensive and scales poorly to large graphs, we apply it only to *NetHEPT* and *DBLP*.

5.1.3 Parameter Settings. In our experiments, we adopt the DIC model under a widely-used setting, where the propagation probability of each edge (u, v) is set as $p_{(u,v)} = 1/InDeg(v)$, with $InDeg(v)$ denoting the in-degree of node v. Our methods can be readily extended to other IC variants or alternative diffusion models with submodular objectives, as supported by many other works on IM variants [9, 23, 24, 50]. We set the seed set size $k = 100$, the error tolerance $\varepsilon = 0.1$, and the failure probability $\delta = 0.05$ by default. For simplicity, we conduct our experiments under the linear aggregation fairness setting. The proposed framework can be readily extended to the welfare-based setting using the techniques introduced in [45]. Since the fairness parameter $\boldsymbol{\alpha}_q$ plays a crucial role in our setting, as it reflects the desired fairness across communities, we assign smaller values of $\alpha_{q,j}$ to larger communities $C_{q,j}$, and larger values to smaller communities. Specifically, we use the following three-level assignment strategy for setting $\alpha_{q,j}$: (1) for communities with size smaller than $n/5$, $\alpha_{q,j}$ is sampled from $U(0, 1)$; (2) for communities with size in the range $[n/5, n/2)$, $\alpha_{q,j}$ is sampled from $U(0, 0.1)$; (3) for communities with size no smaller than $n/2$, $\alpha_{q,j}$ is sampled from $U(0, 0.01)$, where $U(a, b)$ denotes the uniform distribution over the interval $[a, b]$. At the start of each algorithm execution, we run a general FIM algorithm to obtain S_q^g, estimate $\sigma_q(S_q^g)$, and approximate the intractable $\sigma_q(S_q^\circ)$ by scaling the estimate with $1/(1 - 1/e)$ for each $q \in [l]$. This ensures that the scaled estimate is no smaller than the true optimum. This preprocessing step is excluded from the runtime analysis, as all algorithms incur the same expected cost for it.



(a) NetHEPT (b) Amazon (c) DBLP (d) Youtube (e) Wiki (f) LiveJournal

Fig. 3. The objective ρ vs. the seed set size k on different real-world datasets.



(a) NetHEPT (b) Amazon (c) DBLP (d) Youtube (e) Wiki (f) LiveJournal

Fig. 4. The runtime t vs. the seed set size k on different real-world datasets.

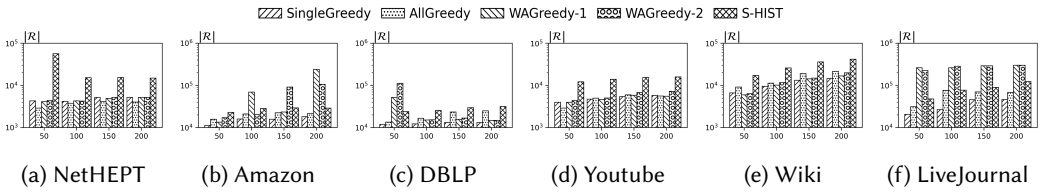
5.1.4 Baseline Methods. Following the baseline design in [24], we employ four baseline algorithms to compare with S-HIST.

- *SingleGreedy:* A simple greedy hill-climbing algorithm that iteratively selects the node with the largest marginal gain in the objective function to add to the seed set. To avoid the high computational cost of MC-simulation-based approaches, we adapt HIST by modifying its objective function, allowing SingleGreedy to maximize the objective based on the coverage of G-RR sets. However, because the objective is non-submodular, this algorithm has no constant-factor approximation guarantee and is used only as a heuristic. Nevertheless, we still force the algorithm to run until the target approximation ratio of $(1-1/e-\varepsilon)$ is achieved.
- *AllGreedy:* For each community partition, we solve the corresponding FIM subproblem by modified HIST to obtain a saturated seed set as a candidate. We then select the candidate seed set that achieves the largest value of the original objective function ρ as the final output. While this algorithm achieves an approximation ratio for each FIM subproblem, it remains a heuristic for RFIM because it does not directly optimize the RFIM objective.
- *WAGreedy-1:* An abbreviation for Weighted Average Greedy 1. It optimizes the CC-DIM objective [21], i.e., the weighted sum of fair influence across all partitions and communities. We use uniform community weights. We implement it using the G-HIST algorithm for CC-DIM. To avoid confusion with SG-HIST, we refer to this algorithm as WAGreedy-1 throughout our experiments.
- *WAGreedy-2:* An abbreviation for Weighted Average Greedy 2. It is identical to WAGreedy-1 but uses different weights, i.e., each community is assigned a weight equal to its fairness parameter. This weighting strategy aligns the optimization with the prescribed fairness priorities.

To ensure a fair comparison with SG-HIST, all baselines are allocated a seed budget of βk. By Algorithm 1, line 1, the saturation factor satisfies $\beta \propto \log(l/\varepsilon)$. For instance, with $k = 100$, $\varepsilon = 0.1$, and $l = 3$, we have $\beta k = 520$. For each configuration of k, ε, and l, the corresponding βk values are listed in the appendix.

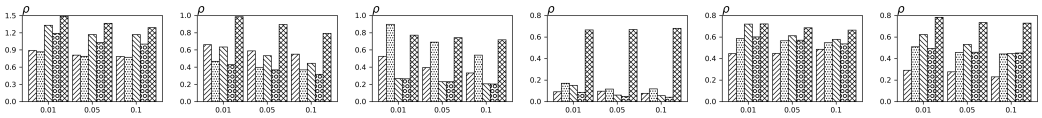
5.2 Results on Real-World Datasets

This section reports experimental results on real-world datasets.



(a) NetHEPT (b) Amazon (c) DBLP (d) Youtube (e) Wiki (f) LiveJournal

Fig. 5. The total number of G-RR sets $|\mathcal{R}|$ vs. the seed set size k on different real-world datasets.



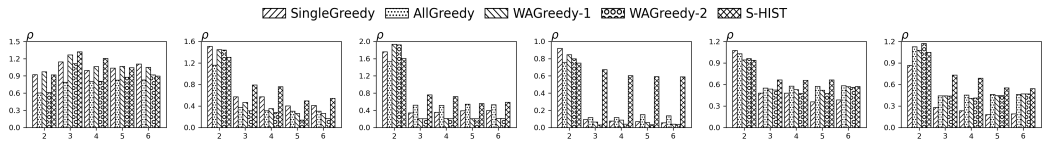
(a) NetHEPT (b) Amazon (c) DBLP (d) Youtube (e) Wiki (f) LiveJournal

Fig. 6. The objective ρ vs. the error tolerance ε on different real-world datasets.

5.2.1 Effect of Seed Set Size. We set $k \in \{50, 100, 150, 200\}$. Fig. 3 shows that S-HIST consistently achieves higher objective ρ across all datasets and seed set sizes, except on *DBLP* when $k = 150$. Notably, S-HIST outperforms the baselines on certain datasets (e.g., *Youtube*), while the other baselines perform competitively on others (e.g., *Wiki*). This variation can be attributed to differences in network topology, community structure, and fairness parameter settings. As RFIM is specifically designed to handle such variations, these results further highlight the robustness of S-HIST across diverse network and fairness conditions. Fig. 4 presents the runtime comparison. S-HIST remains practical in terms of efficiency, with its runtime at most within one order of magnitude of the fastest baseline, SingleGreedy, which is still acceptable in practice. Notably, AllGreedy exhibits higher runtime, as it has to evaluate the performance on RFIM of each candidate seed set selected by optimizing the FIM subproblem on each partition, for which we employ a simple MC-simulation-based method. Fig. 5 presents a comparison of the number of G-RR sets. Consistent with the runtime results, S-HIST typically samples at most one order of magnitude more G-RR sets than the most sample-efficient baseline, SingleGreedy. This behavior is expected and acceptable.

5.2.2 Effect of Error Tolerance. We set $\varepsilon \in \{0.01, 0.05, 0.1\}$. Similar to the results above, Fig. 6 shows that our proposed S-HIST consistently outperforms the baseline algorithms in maximizing the objective ρ across all datasets, except on *DBLP* with $\varepsilon = 0.01$. Once again, the four baselines exhibit limited robustness under variations in both graph structure and community partitions. Due to space constraints, the runtime and G-RR set comparisons, which are analogous to Fig. 4 and Fig. 5, are provided in the appendix. Notably, the observed runtime does not exhibit the ε^6 dependence in the denominator of the complexity bound discussed in Section 4.3. This discrepancy arises because the analysis assumes $OPT = \Omega(\varepsilon)$, which is often overly conservative in practice. Empirically, we typically have $OPT \gg O(\varepsilon)$, then the ε^6 term reduces to ε^2, in line with general IM complexity and with our experimental results.

5.2.3 Effect of the Number of Partitions. Designed to mitigate allocation inequality across multiple partitions, S-HIST is evaluated as the number of partitions increases. Recall that a partition in RFIM is defined by the community assignment and its associated vector of fairness parameters. Together these two components define a single-partition FIM subproblem, as discussed in Section 1 and Section 3.2. To construct complementary yet plausible partitions, we fix the community assignment



(a) NetHEPT (b) Amazon (c) DBLP (d) Youtube (e) Wiki (f) LiveJournal

Fig. 7. The objective ρ vs. the number of partitions $l = |\mathcal{P}|$ on different real-world datasets.

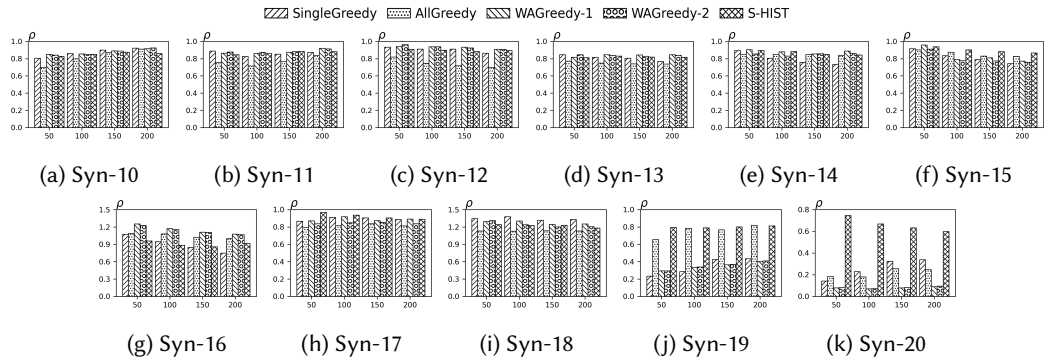
and vary the fairness parameters. We deliberately refrain from introducing additional community detection methods or further adjusting their hyperparameters, as this choice simplifies the setting. Meanwhile, different detection methods on the same graph may yield similar community structures, thereby inducing similar FIM subproblems, especially under our three-level assignment strategy for fairness parameters. Purely random partitions are also undesirable. For example, assigning community labels uniformly at random spreads each community nearly uniformly across the graph, so the objective tends to prioritize globally influential nodes regardless of the fairness parameters. Under such a construction, the resulting FIM subproblem effectively reduces to standard IM.

Accordingly, we adopt the following partition augmentation strategy. Given a base partition, we keep each node's community label unchanged. For each community $C_{q,j}$, we replace its fairness parameter $\alpha_{q,j}$ with its complement $1-\alpha_{q,j}$ with probability 0.75, and otherwise leave $\alpha_{q,j}$ unchanged. For example, if a community $C_{q,j}$ has $\alpha_{q,j} = 0.2$, then with probability 0.75 the parameter becomes 0.8, and with probability 0.25 it remains 0.2. We refer to the resulting partition as the *inverse* of the base partition. This strategy emphasizes communities that were assigned lower fairness weights in the base partition, while retaining randomness to avoid a perfectly mirrored design. For all datasets, we vary $l = |\mathcal{P}| \in \{2, 3, 4, 5, 6\}$. For $l = 2$ we use *Ground-Truth* and its inverse. For $l = 3$ we use *Ground-Truth*, *Leiden*, and *Label Propagation*. For $l = 4$ we extend the $l = 3$ selection by adding the inverse of *Ground-Truth*. For $l = 5$ we extend the $l = 4$ selection by adding the inverse of *Leiden*. For $l = 6$ we extend the $l = 5$ selection by adding the inverse of *Label Propagation*. This construction also applies to all synthetic datasets introduced in Section 5.3. The only exception is *NetHEPT*, which does not have a *Ground-Truth* partition. For $l = 2$ we use *Leiden* and its inverse. For $l \geq 3$ we fix the order {*Leiden*, *Label Propagation*, *Spectral Clustering*, inverse of *Leiden*, inverse of *Label Propagation*, inverse of *Spectral Clustering*} and take the first l entries.

The results are shown in Fig. 7. All methods perform well when $l = 2$. However, as l increases, the objective ρ generally decreases, and only S-HIST consistently maintains high performance for larger l, except for *NetHEPT*, which lacks a *Ground-Truth* partition. This trend is natural because the worst-case ratio becomes harder to sustain as the number of candidate partitions increases. We defer the results on runtime and number of G-RR sets to the appendix. Simply put, both metrics increase as l grows, and the cross-method comparisons are consistent with those in Section 5.2.1.

5.3 Results on Synthetic Datasets

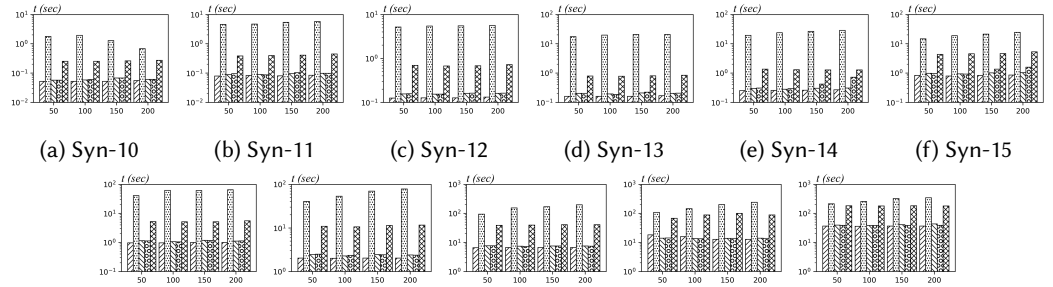
To evaluate the scalability of S-HIST, we use the LFR benchmark [58] to generate synthetic graphs with ground-truth non-overlapping community labels. The graph sizes are set to 2^{10}, 2^{11}, …, 2^{20} nodes, and we refer to each synthetic graph as *Syn-10*, *Syn-11*, and so on, corresponding to its exponent. Since RFIM requires multiple community partitions as input, we apply *Leiden*, *Label Propagation*, and *Spectral Clustering* to obtain diverse partitions. Due to the high complexity of *Spectral Clustering*, we apply it only to graphs up to *Syn-14*. For *Label Propagation*, we assign each node a distinct initial label. For fairness parameter settings, we adopt the same three-level assignment strategy as used in the real-world networks.



(a) Syn-10 (b) Syn-11 (c) Syn-12 (d) Syn-13 (e) Syn-14 (f) Syn-15

(g) Syn-16 (h) Syn-17 (i) Syn-18 (j) Syn-19 (k) Syn-20

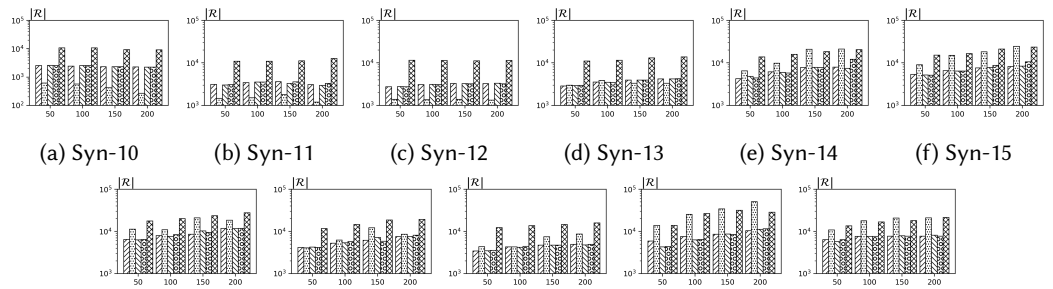
Fig. 8. The objective ρ vs. the seed set size k on different synthetic datasets.



(a) Syn-10 (b) Syn-11 (c) Syn-12 (d) Syn-13 (e) Syn-14 (f) Syn-15

(g) Syn-16 (h) Syn-17 (i) Syn-18 (j) Syn-19 (k) Syn-20

Fig. 9. The runtime t vs. the seed set size k on different synthetic datasets.



(a) Syn-10 (b) Syn-11 (c) Syn-12 (d) Syn-13 (e) Syn-14 (f) Syn-15

(g) Syn-16 (h) Syn-17 (i) Syn-18 (j) Syn-19 (k) Syn-20

Fig. 10. The total number of G-RR sets $|\mathcal{R}|$ vs. the seed set size k on different synthetic datasets.

Mirroring the experiments on real-world datasets, we sweep k, ε, and l in turn, holding the remaining parameters fixed for each sweep. We use the same partition augmentation strategy described in Section 5.2.3. We present results as functions of k for three metrics, i.e., ρ, t, and $|\mathcal{R}|$ (see Fig. 8, Fig. 9, and Fig. 10, respectively). Overall, excluding *Syn-19* and *Syn-20*, the synthetic graphs yield comparable objective values ρ across the five algorithms. In other words, all algorithms perform reasonably well on the synthetic graphs. This observation can be explained by the fact

that LFR-benchmark graphs are constructed with clear community structures and relatively sparse inter-community edges. As a result, community detection methods tend to recover partitions that closely align with the "ground-truth" communities embedded during graph generation. In contrast, in real-world networks, the true community structures often differ from the outputs of standard community detection algorithms, leading to greater variability in algorithm performance. The observations for varying ε and l are similar to those on the real-world datasets, and the cross-method comparisons are consistent with Fig. 8, Fig. 9, and Fig. 10. We therefore defer these results to the appendix.

In summary, the results on the synthetic datasets reveal practical limitations of S-HIST relative to the baselines. In some cases, S-HIST attains a slightly lower objective value ρ than the strongest baseline among the four implemented baselines, and its runtime is up to one order of magnitude higher than the fastest baseline (typically *SingleGreedy*). Nevertheless, its stability and robustness across settings make it generally the most suitable choice for RFIM.

6 Conclusion and Future Work

In this paper, we formulate the RFIM problem to address the challenge of handling multiple possible community partitions in FIM. We show that approximating the RFIM problem is strongly NP-hard due to its non-submodularity. We propose the S-HIST algorithm, which allows a saturated seed set size to achieve a $(1 - 1/e - \varepsilon/OPT)$-approximation ratio with high probability. To efficiently maximize the surrogate objective in S-HIST, we further develop the SG-HIST algorithm as its core subroutine. We conduct extensive experiments on real-world and synthetic datasets, demonstrating that S-HIST achieves robust and stable performance with competitive effectiveness and efficiency compared to several baselines.

Future work may extend in several complementary directions. One is to combine seed minimization with the optimization of the submodular surrogate used in the S-HIST subroutine. Another is to design efficient heuristics for RFIM. It is also worthwhile to evaluate a broader set of fairness metrics for RFIM, including application-specific ones. Lastly, improving community detection methods for FIM offers a complementary perspective to RFIM, focusing on enhancing partition quality rather than adapting to imperfect ones.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number JP22H00533, Japan.

References

[1] Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. 2019. Learning Optimal and Fair Decision Trees for Non-Discriminative Decision-Making. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 1418–1426. https://doi.org/10.1609/aaai.v33i01.33011418

[2] Junaid Ali, Mahmoudreza Babaei, Abhijnan Chakraborty, Baharan Mirzasoleiman, Krishna P. Gummadi, and Adish Singla. 2023. On the Fairness of Time-Critical Influence Maximization in Social Networks. *IEEE Transactions on Knowledge and Data Engineering* 35, 3 (March 2023), 2875–2886. https://doi.org/10.1109/TKDE.2021.3120561

[3] Nima Anari, Nika Haghtalab, Seffi Naor, Sebastian Pokutta, Mohit Singh, and Alfredo Torrico. 2019. Structured Robust Submodular Maximization: Offline and Online Algorithms. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 89)*, Kamalika Chaudhuri and Masashi Sugiyama (Eds.). PMLR, 3128–3137. https://proceedings.mlr.press/v89/anari19a.html

[4] Aharon Ben-Tal and Arkadi Nemirovski. 2002. Robust optimization - methodology and applications. *Math. Program.* 92, 3 (2002), 453–480. https://doi.org/10.1007/S101070100286

[5] Shishir Bharathi, David Kempe, and Mahyar Salek. 2007. Competitive Influence Maximization in Social Networks. In *Internet and Network Economics*, Xiaotie Deng and Fan Chung Graham (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 306–311.

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (oct 2008), P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

[7] 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* (Portland, Oregon) *(SODA '14)*. Society for Industrial and Applied Mathematics, USA, 946–957.

[8] Robert S. Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. 2017. Robust Optimization for Non-Convex Objectives. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/10c66082c124f8afe3df4886f5e516e0-Paper.pdf

[9] Wei Chen, Tian Lin, Zihan Tan, Mingfei Zhao, and Xuren Zhou. 2016. Robust Influence Maximization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 795–804. https://doi.org/10.1145/2939672.2939745

[10] 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* (Washington, DC, USA) *(KDD '10)*. Association for Computing Machinery, New York, NY, USA, 1029–1038. https://doi.org/10.1145/1835804.1835934

[11] 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* (Paris, France) *(KDD '09)*. Association for Computing Machinery, New York, NY, USA, 199–208. https://doi.org/10.1145/1557019.1557047

[12] 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*. 88–97. https://doi.org/10.1109/ICDM.2010.118

[13] Xiaolong Chen, Yifan Song, and Jing Tang. 2024. Link Recommendation to Augment Influence Diffusion with Provable Guarantees. In *Proceedings of the ACM Web Conference 2024* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 2509–2518. https://doi.org/10.1145/3589334.3645521

[14] Xiaolong Chen and Jing Tang. 2025. Scalable Link Recommendation for Influence Maximization. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1* (Toronto ON, Canada) *(KDD '25)*. Association for Computing Machinery, New York, NY, USA, 130–141. https://doi.org/10.1145/3690624.3709190

[15] Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F. Werneck. 2014. Sketch-based Influence Maximization and Computation: Scaling up with Guarantees. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (Shanghai, China) *(CIKM '14)*. Association for Computing Machinery, New York, NY, USA, 629–638. https://doi.org/10.1145/2661829.2662077

[16] Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. 2019. Recommending links through influence maximization. *Theoretical Computer Science* 764 (2019), 30–41. https://doi.org/10.1016/j.tcs.2018.01.017 Selected papers of ICTCS 2016 (The Italian Conference on Theoretical Computer Science (ICTCS).

[17] Chen Feng, Xingguang Chen, Qintian Guo, Fangyuan Zhang, and Sibo Wang. 2024. Efficient Approximation Algorithms for Minimum Cost Seed Selection with Probabilistic Coverage Guarantee. *Proc. ACM Manag. Data* 2, 4, Article 197 (Sept. 2024), 26 pages. https://doi.org/10.1145/3677133

[18] Yuting Feng, Ankitkumar Patel, Bogdan Cautis, and Hossein Vahabi. 2023. Influence Maximization with Fairness at Scale. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 4046–4055. https://doi.org/10.1145/3580305.3599847

[19] Benjamin Fish, Ashkan Bashardoust, Danah Boyd, Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2019. Gaps in Information Access in Social Networks?. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 480–490. https://doi.org/10.1145/3308558.3313680

[20] Santo Fortunato. 2010. Community detection in graphs. *Physics Reports* 486, 3 (2010), 75–174. https://doi.org/10.1016/j.physrep.2009.11.002

[21] Jianxiong Guo, Qiufen Ni, Weili Wu, and Ding-Zhu Du. 2024. Composite Community-Aware Diversified Influence Maximization With Efficient Approximation. *IEEE/ACM Transactions on Networking* 32, 2 (April 2024), 1584–1599. https://doi.org/10.1109/TNET.2023.3321870

[22] Qintian Guo, Sibo Wang, Zhewei Wei, and Ming Chen. 2020. Influence Maximization Revisited: Efficient Reverse Reachable Set Generation with Bound Tightened. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 2167–2181. https://doi.org/10.1145/3318464.3389740

[23] Qintian Guo, Sibo Wang, Zhewei Wei, Wenqing Lin, and Jing Tang. 2022. Influence Maximization Revisited: Efficient Sampling with Bound Tightened. *ACM Trans. Database Syst.* 47, 3, Article 12 (Aug. 2022), 45 pages. https://doi.org/10.1145/3533817

[24] Xinran He and David Kempe. 2016. Robust Influence Maximization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 885–894. https://doi.org/10.1145/2939672.2939760

[25] Xinran He and David Kempe. 2018. Stability and Robustness in Influence Maximization. *ACM Trans. Knowl. Discov. Data* 12, 6, Article 66 (Aug. 2018), 34 pages. https://doi.org/10.1145/3233227

[26] Dimitris Kalimeris, Gal Kaplun, and Yaron Singer. 2019. Robust Influence Maximization for Hyperparametric Models. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 3192–3200. https://proceedings.mlr.press/v97/kalimeris19a.html

[27] 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* (Washington, D.C.) *(KDD '03)*. Association for Computing Machinery, New York, NY, USA, 137–146. https://doi.org/10.1145/956750.956769

[28] David Kempe, Jon Kleinberg, and Éva Tardos. 2015. Maximizing the Spread of Influence through a Social Network. *Theory of Computing* 11, 4 (2015), 105–147. https://doi.org/10.4086/toc.2015.v011a004

[29] Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. 2008. Robust Submodular Observation Selection. *Journal of Machine Learning Research* 9, 12 (2008).

[30] Siyu Lei, Silviu Maniu, Luyi Mo, Reynold Cheng, and Pierre Senellart. 2015. Online Influence Maximization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia) *(KDD '15)*. Association for Computing Machinery, New York, NY, USA, 645–654. https://doi.org/10.1145/2783258.2783271

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

[32] Shuai Li, Fang Kong, Kejie Tang, Qizhi Li, and Wei Chen. 2020. Online influence maximization under linear threshold model. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 101, 13 pages.

[33] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. 2018. Influence Maximization on Social Graphs: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (Oct 2018), 1852–1872. https://doi.org/10.1109/TKDE.2018.2807843

[34] Yandi Li, Haobo Gao, Yunxuan Gao, Jianxiong Guo, and Weili Wu. 2023. A Survey on Influence Maximization: From an ML-Based Combinatorial Optimization. *ACM Trans. Knowl. Discov. Data* 17, 9, Article 133 (July 2023), 50 pages. https://doi.org/10.1145/3604559

[35] Wei Lu, Wei Chen, and Laks V. S. Lakshmanan. 2015. From competition to complementarity: comparative influence diffusion and maximization. *Proc. VLDB Endow.* 9, 2 (Oct. 2015), 60–71. https://doi.org/10.14778/2850578.2850581

[36] 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.

[37] Hung T. Nguyen, Tri P. Nguyen, Tam N. Vu, and Thang N. Dinh. 2017. Outward Influence and Cascade Size Estimation in Billion-scale Networks. *Proc. ACM Meas. Anal. Comput. Syst.* 1, 1, Article 20 (June 2017), 30 pages. https://doi.org/10.1145/3084457

[38] 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* (San Francisco, California, USA) *(SIGMOD '16)*. Association for Computing Machinery, New York, NY, USA, 695–710. https://doi.org/10.1145/2882903.2915207

[39] Binghui Peng. 2021. Dynamic influence maximization. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NIPS '21)*. Curran Associates Inc., Red Hook, NY, USA, Article 820, 14 pages.

[40] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 76 (Sep 2007), 036106. Issue 3. https://doi.org/10.1103/PhysRevE.76.036106

[41] Aida Rahmattalabi, Shahin Jabbari, Himabindu Lakkaraju, Phebe Vayanos, Max Izenberg, Ryan Brown, Eric Rice, and Milind Tambe. 2021. Fair Influence Maximization: a Welfare Optimization Approach. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 13 (May 2021), 11630–11638. https://doi.org/10.1609/aaai.v35i13.17383

[42] John Rawls. 2017. A theory of justice. In *Applied ethics*. Routledge, 21–29.

[43] Behnam Razaghi, Mehdy Roayaei, and Nasrollah Moghadam Charkari. 2023. On the Group-Fairness-Aware Influence Maximization in Social Networks. *IEEE Transactions on Computational Social Systems* 10, 6 (Dec 2023), 3406–3414.

https://doi.org/10.1109/TCSS.2022.3198096

[44] Xiaobin Rui, Zhixiao Wang, Hao Peng, Wei Chen, and Philip S. Yu. 2025. A Scalable Algorithm for Fair Influence Maximization with Unbiased Estimator . *IEEE Transactions on Knowledge & Data Engineering* 01 (April 2025), 1–15. https://doi.org/10.1109/TKDE.2025.3564283

[45] Xiaobin Rui, Zhixiao Wang, Jiayu Zhao, Lichao Sun, and Wei Chen. 2023. Scalable fair influence maximization. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 2911, 17 pages.

[46] Arkaprava Saha, Bogdan Cautis, Xiaokui Xiao, and Laks V. S. Lakshmanan. 2025. Hyperparametric Robust and Dynamic Influence Maximization. *Proceedings of the AAAI Conference on Artificial Intelligence* 39, 12 (Apr. 2025), 12497–12505. https://doi.org/10.1609/aaai.v39i12.33362

[47] Ana-Andreea Stoica, Jessy Xinyi Han, and Augustin Chaintreau. 2020. Seeding Network Influence in Biased Networks and the Benefits of Diversity. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2089–2098. https://doi.org/10.1145/3366423.3380275

[48] Jing Tang, Keke Huang, Xiaokui Xiao, Laks V.S. Lakshmanan, Xueyan Tang, Aixin Sun, and Andrew Lim. 2019. Efficient Approximation Algorithms for Adaptive Seed Minimization. In *Proceedings of the 2019 International Conference on Management of Data* (Amsterdam, Netherlands) *(SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 1096–1113. https://doi.org/10.1145/3299869.3319881

[49] 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* (Houston, TX, USA) *(SIGMOD '18)*. Association for Computing Machinery, New York, NY, USA, 991–1005. https://doi.org/10.1145/3183713.3183749

[50] 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* (Melbourne, Victoria, Australia) *(SIGMOD '15)*. Association for Computing Machinery, New York, NY, USA, 1539–1554. https://doi.org/10.1145/2723372.2723734

[51] 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* (Snowbird, Utah, USA) *(SIGMOD '14)*. Association for Computing Machinery, New York, NY, USA, 75–86. https://doi.org/10.1145/2588555.2593670

[52] Guangmo Tong, Weili Wu, Shaojie Tang, and Ding-Zhu Du. 2017. Adaptive Influence Maximization in Dynamic Social Networks. *IEEE/ACM Transactions on Networking* 25, 1 (Feb 2017), 112–125. https://doi.org/10.1109/TNET.2016.2563397

[53] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports* 9, 1 (2019), 1–12. https://doi.org/10.1038/s41598-019-41695-z

[54] Alan Tsang, Bryan Wilder, Eric Rice, Milind Tambe, and Yair Zick. 2019. Group-Fairness in Influence Maximization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. 5997–6005. https://doi.org/10.24963/ijcai.2019/831

[55] Ulrike von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing* 17, 4 (2007), 395–416. https://doi.org/10.1007/S11222-007-9033-Z

[56] Xiaoyang Wang, Ying Zhang, Wenjie Zhang, Xuemin Lin, and Chen Chen. 2017. Bring Order into the Samples: A Novel Scalable Method for Influence Maximization. *IEEE Transactions on Knowledge and Data Engineering* 29, 2 (Feb 2017), 243–256. https://doi.org/10.1109/TKDE.2016.2624734

[57] Zhixiao Wang, Jiayu Zhao, Chengcheng Sun, Xiaobin Rui, and Philip S. Yu. 2024. A General Concave Fairness Framework for Influence Maximization Based on Poverty Reward. *ACM Trans. Knowl. Discov. Data* 19, 1, Article 14 (Dec. 2024), 23 pages. https://doi.org/10.1145/3701737

[58] Zhao Yang, Juan I. Perotti, and Claudio J. Tessone. 2017. Hierarchical benchmark graphs for testing community detection algorithms. *Physical Review E* 96 (Nov 2017), 052311. Issue 5. https://doi.org/10.1103/PhysRevE.96.052311

Received July 2025; revised October 2025; accepted November 2025