Multi-Round Influence Maximization

Lichao Sun[1], Weiran Huang[2], Philip S. Yu[1,2], Wei Chen[3,*]
[1]University of Illinois at Chicago, [2]Tsinghua University, [3]Microsoft Research
lsun29@uic.edu, huang.inbox@outlook.com, psyu@uic.edu, weic@microsoft.com

ABSTRACT

In this paper, we study the Multi-Round Influence Maximization (MRIM) problem, where influence propagates in multiple rounds independently from possibly different seed sets, and the goal is to select seeds for each round to maximize the expected number of nodes that are activated in at least one round. MRIM problem models the viral marketing scenarios in which advertisers conduct multiple rounds of viral marketing to promote one product. We consider two different settings: 1) the non-adaptive MRIM, where the advertiser needs to determine the seed sets for all rounds at the very beginning, and 2) the adaptive MRIM, where the advertiser can select seed sets adaptively based on the propagation results in the previous rounds. For the non-adaptive setting, we design two algorithms that exhibit an interesting tradeoff between efficiency and effectiveness: a cross-round greedy algorithm that selects seeds at a global level and achieves $1/2-\varepsilon$ approximation ratio, and a within-round greedy algorithm that selects seeds round by round and achieves $1-e^{-(1-1/e)}-\varepsilon \approx 0.46-\varepsilon$ approximation ratio but saves running time by a factor related to the number of rounds. For the adaptive setting, we design an adaptive algorithm that guarantees $1-e^{-(1-1/e)}-\varepsilon$ approximation to the adaptive optimal solution. In all cases, we further design scalable algorithms based on the reverse influence sampling approach and achieve near-linear running time. We conduct experiments on several real-world networks and demonstrate that our algorithms are effective for the MRIM task.

CCS CONCEPTS

• **Information systems** → *Social advertising*; *Social networks*; • **Theory of computation** → *Probabilistic computation*; *Submodular optimization and polymatroids*;

KEYWORDS

Influence maximization, triggering model, greedy algorithm

ACM Reference Format:
Lichao Sun[1], Weiran Huang[2], Philip S. Yu[1,2], Wei Chen[3,]. 2018. Multi-Round Influence Maximization . In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3220101

*Corresponding author

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3220101

1 INTRODUCTION

Most companies need to advertise their products or brands on social networks, through paying for influential people (seed nodes) on Twitter, with the hope that they can promote the products to their followers [2]. The objective is to find a set of most influential people with limited budget for the best marketing effect. Influence maximization (IM) is the optimization problem of finding a small set of most influential nodes in a social network that generates the largest influence spread. It models viral marketing scenario in social networks [12, 20, 27], and can also be applied to cascade detection [22], rumor control [18], etc. The standard IM problem and a number of its variants has been extensively studied (c.f. [7]). In most formulations, the IM is formulated as a one-shot task: the seed set is selected by the algorithm at the beginning, and one propagation pass from the seed set activates a subset of nodes in the network. The objective is to maximize the expected number of activated nodes in this one propagation pass. However, in many practical viral marketing scenarios, an advertiser's viral marketing campaign may contain multiple rounds to promote one product. Each round may be initiated from a different set of influential nodes. The advertiser would like to maximize the total number of users adopting the product over all rounds.

We model the above scenario by the multi-round diffusion model and multi-round influence maximization (MRIM) task. We consider the entire process over T rounds. In each round t, an independent diffusion is carried out starting from seed set S_t, and a random set of nodes, $I(S_t)$, is activated. Then the total influence spread over T rounds given seed sets $S_t, \ldots, S_T$, $\rho(S_1, \ldots, S_T)$, is the expected total number of activated nodes while considering all rounds together, namely $\rho(S_1, \ldots, S_T) = \mathbb{E}[|\bigcup_{t=1}^{T} I(S_t)|]$. Note that a node activated in a previous round may be activated again and propagate influence to other nodes in a new round, but it will not be counted again in the final influence spread. The MRIM task is to find seed sets $S_1, \ldots, S_T$, each of which has at most k nodes, so that the final influence spread $\rho(S_1, \ldots, S_T)$ is maximized.

The MRIM task possesses some unique features different from the classical IM task. For example, in the classical IM, it makes no sense to select one seed node multiple times, but for MRIM, it may be desirable to select an influential node in multiple rounds to generate more influence. Moreover, it enables adaptive strategies where the advertiser to adaptive select seeds in the next round based on the propagation results of previous rounds.

We study both the non-adaptive and adaptive versions of MRIM under the Multi-Round Triggering (MRT) model. For the non-adaptive MRIM problem, we design two algorithms that exhibit an interesting trade-off between efficiency and effectiveness. The first cross-round greedy algorithm selects seeds globally cross different rounds, and achieves an approximation ratio of $1/2-\varepsilon$, for any $\varepsilon > 0$. The second within-round greedy algorithm selects seeds round by round and achieves an approximation ratio of $1-e^{-(1-1/e)}-\varepsilon \approx 0.46-\varepsilon$.

The higher approximation ratio enjoyed by the cross-round greedy algorithm is achieved by investigating seed candidates in all rounds together in each greedy step, and thus incurs a higher running time cost at a factor proportional to the number of rounds. For the adaptive MRIM problem, we rigorously formulate the problem according to the adaptive optimization framework specified in [14]. We show that our formulation satisfies the *adaptive submodularity* defined in [14]. Based on the adaptive submodularity, we design the AdaGreedy algorithm that achieves $1 - e^{-(1-1/e)} - \varepsilon$ approximation to the optimal adaptive policy.

For both the non-adaptive and adaptive cases, we greatly improve the scalability by incorporating the state-of-the-art reverse influence sampling (RIS) approach [4, 30].In each case, the RIS method needs to be carefully revised to accommodate to the multi-round or adaptive situation. In all cases, we show that our scalable algorithms could achieve near-linear running time with respect to the network size, greatly improving the corresponding Monte Carlo greedy algorithms.

To demonstrate the effectiveness of our algorithms, we conduct experiments on real-world social networks, with both synthesized and learned influence parameters. Our experimental results demonstrate that our algorithms are more effective than baselines, and our scalable algorithms run in orders of magnitude faster than their Monte Carlo greedy counterparts while keeping the influence spread at the same level. The results also show some interesting findings, such as the non-adaptive cross-round algorithm could achieve competitive influence spread as the adaptive greedy algorithm. This may suggest that in practice one may need to consider whether spending the cost of collecting feedbacks and doing near-term adaptive strategies based on feedbacks, or spending the up front cost to do more global planning, and it opens new directions for further investigations.

To summarize, our contributions are: (a) proposing the study of both non-adaptive and adaptive MRIM problems; (b) proposing non-adaptive and adaptive greedy algorithms and showing their trade-offs; (c) designing scalable algorithms in both non-adaptive and adaptive settings; and (d) conducting experiments on real-world networks to demonstrate the effectiveness and the scalability of our proposed algorithms. Due to space constraints, all proofs and supplementary materials are shown in the extended version [29].

1.1 Related Work

Influence maximization is first studied by Domingos and Richardson [12, 27], and then formulated as a discrete optimization problem by Kempe et al. [20], who also formulate the independent cascade model, the linear threshold model, the triggering model, and provide a greedy approximation algorithm based on submodularity. Since then, a significant number of papers studies improving the efficiency and scalability of influence maximization algorithms [4, 9, 11, 19, 30, 32]. At this front, the state of the art is the reverse influence sampling (RIS) approach [4, 30], and the IMM algorithm of [30] is among the most competitive ones so far. Our scalable algorithms are based on IMM, but require careful redesigns in the reverse sampling method. Other studies look into different variants, such as community, competitive and complementary influence maximization [6, 16, 18, 24], adoption maximization [3], robust influence maximization [8, 17], etc.

Another related work is adaptive seeding [28], which uses the first-stage nodes and their accessible neighbors together as the seed set to maximize the influence, and is quite different from ours. In terms of the multi-round diffusion model and influence maximization, Lin et al. [23] focus on the multi-party influence maximization where there must be at least two parties to compete in networks. Lei et al. [21] use the same formulation of the multi-round diffusion model and the influence maximization objective as in our paper. They focus on the online learning aspect of learning edge probabilities, while we study the offline non-adaptive and adaptive maximization problem when the edge probabilities are known. Without a rigorous study of such offline problems, it is very difficult to assess the performance of online learning algorithms, and as a result they could only propose heuristic learning algorithms without any theoretical guarantee. From this perspective, our study fills this important gap by providing a solid theoretical understanding of the offline multi-round influence maximization.

There are also a number of studies on the influence maximization bandit problem [10, 31, 33]. Their formulations also have multi-rounds, but each round has a separate influence maximization instance, and the total reward is a simple count of activated nodes in all rounds, so one node activated in multiple rounds will be counted multiple times. This makes it qualitatively different from our formulation. Moreover, these study focus on the online learning aspects of such repeated influence maximization tasks.

The adaptive MRIM study follows the adaptive optimization framework defined by Golovin and Krause [14]. They also study adaptive influence maximization as an application, but the adaptation is at per-node level: seeds are selected one by one. Later seed can be selected based on the activation results from the earlier seeds, but the earlier seeds would not help propagation again for later seeds. This makes it different from our multi-round model.

2 MODEL AND PROBLEM DEFINITION

2.1 Multi-Round Diffusion Model

In this paper, we focus on the well-studied *triggering model* [20] as the basic diffusion model. A social network is modeled as a directed graph $G = (V, E)$, where V is a finite set of vertices or nodes, and $E \subseteq V \times V$ is the set of directed edges connecting pairs of nodes. The diffusion of information or influence proceeds in discrete time steps $\tau = 0, 1, 2, \ldots$. At time $\tau = 0$, the *seed set* S_0 is selected to be active, and also each node v independently chooses a random *triggering set* $T(v)$ according to some distribution over subsets of its in-neighbors. At each time $\tau \geq 1$, an inactive node v becomes active if at least one node in $T(v)$ is active by time $\tau - 1$. The diffusion process ends when there is no more nodes activated in a time step. We remark that the classical Independent Cascade (IC) model is a special case of the triggering model. In the IC model, every edge $(u, v) \in E$ is associated with a probability $p_{uv} \in [0, 1]$, and p_{uv} is set to zero if $(u, v) \notin E$. Each triggering set $T(v)$ is generated by independently sampling (u, v) with probability p_{uv} and including u in $T(v)$ if the sampling of (u, v) is successful.

The triggering model can be equivalently described as propagations in *live-edge graphs*. Given a class of triggering sets $\{T(v)\}_{v \in V}$, we can construct the live-edge graph $L = (V, E(L))$, where $E(L) = \{(u, v) \mid v \in V, u \in T(v)\}$, and each edge $(u, v) \in L$ is called a *live*

edge. It is easy to see that the propagation in the triggering model is the same as the deterministic propagation in the corresponding live-edge graph L: if A is the set of active nodes at time $\tau - 1$, then all directed out-neighbors of nodes in A will be active by time τ. An important metric in any diffusion model is the *influence spread*, the expected number of active nodes when the propagation from the given seed set S_0 ends, denoted as $\sigma(S_0)$. Let $\Gamma(G, S)$ denote the set of nodes in graph G that can be reached from the node set S. Then, by the above equivalent live-edge graph model, we have $\sigma(S_0) = \mathbb{E}[|\Gamma(L, S_0)|]$, where the expectation is taken over the distribution of live-edge graphs.

A set function $f : V \to \mathbb{R}$ is called *submodular* if for all $S \subseteq O \subseteq V$ and $u \in V \setminus T$, $f(S \cup \{u\}) - f(S) \geq f(O \cup \{u\}) - f(O)$. Intuitively, submodularity characterizes the diminishing return property often occurring in economics and operation research. Moreover, a set function f is called *monotone* if for all $S \subseteq O \subseteq V$, $f(S) \leq f(O)$. It is shown in [20] that influence spread σ for the triggering model is a monotone submodular function . A non-negative monotone submodular function allows a greedy solution to its maximization problem subject to a cardinality constraint, with an approximation ratio $1 - 1/e$, where e is the base of the natural logarithm [26]. This is the technical foundation for most influence maximization tasks.

We are now ready to define the Multi-Round Triggering (MRT) model. The MRT model includes T *independent* rounds of influence diffusions. In each round $t \in [T]$, the diffusion starts from a separate seed set S_t with up to k nodes, and it follows the dynamic in the classical triggering model described previously. Since one node can be repeatedly selected as the seed set in different rounds, to clarify the round, we use pair notation (v, t) to denote a node v in the seed set of round t. We use $\mathcal{S}_t = \{(v, t) \mid v \in S_t\}$ to represent the seed set of round t in the pair notation. Henceforth, we always use the calligraphic $\mathcal{S}$ for sets in the pair notation and the normal S for node sets. By the equivalence between the triggering model and the live-edge graph model, the MRT model can be viewed as T independent propagations in the T live-edge graphs $L_1, L_2, \ldots, L_T$, which are drawn independently from the same distribution based on the triggering model. The total active nodes in T rounds is counted by $\left|\bigcup_{t=1}^{T} \Gamma(L_t, S_t)\right|$, where $\Gamma(L_t, S_t)$ is the set of final active nodes in round t. Given a class of seed set (in pair notation) $\mathcal{S} := \bigcup_{t=1}^{T} \mathcal{S}_t$, the *influence spread* ρ in the MRT model is defined as

$$\rho(\mathcal{S}) = \rho(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_T) = \mathbb{E}\left[\left|\bigcup_{t=1}^{T} \Gamma(L_t, S_t)\right|\right],$$

where the expectation is taken over the distribution of all live-edge graphs $L_1, L_2, \ldots, L_T$.

2.2 Multi-Round Influence Maximization

The classical influence maximization problem is to choose a seed set S_0 of size at most k to maximize the influence spread $\sigma(S_0)$. For the Multi-Round Influence Maximization (MRIM) problem, the goal is to select at most k seed nodes of each round, such that the influence spread in T rounds is maximized. We first introduce its non-adaptive version formally defined as follows.

Definition 2.1. The *non-adaptive* Multi-Round Influence Maximization (MRIM) under the MRT model is the optimization task where the input includes the directed graph $G = (V, E)$, the triggering set distribution for every node in the MRT model, the number of rounds T, and each-round budget k, and the goal is to find T seed sets $\mathcal{S}_1^*, \mathcal{S}_2^*, \ldots, \mathcal{S}_T^*$ with each seed set having at most k nodes, such that the total influence spread is maximized, i.e.,

$$\mathcal{S}^* = \mathcal{S}_1^* \cup \mathcal{S}_2^* \cdots \cup \mathcal{S}_T^* = \operatorname*{argmax}_{\mathcal{S}:\, |S_t| \leq k, \forall t \in [T]} \rho(\mathcal{S}).$$

The non-adaptiveness refers to the definition that one needs to find T seed sets all at once before the propagation starts. In practice, one may be able to observe the propagation results in previous rounds and select the seed set for the next round based on the previous results to increase the influence spread. This leads to the adaptive version. To formulate the *adaptive* MRIM requires the setup of the adaptive optimization framework, and we defer to Section 4.1 for its formal definition.

Note that as the classical influence maximization is NP-hard and is a special case of MRIM with $T = 1$, both the non-adaptive and adaptive versions of MRIM are NP-hard.

3 NON-ADAPTIVE MRIM

Let $\mathcal{V}_t = \{(v, t) \mid v \in V\}$ be the set of all possible nodes in round t (e.g., $\mathcal{S}_t \subseteq \mathcal{V}_t$), and $\mathcal{V} := \bigcup_{t=1}^{T} \mathcal{V}_t$. We first show that the influence spread function ρ for the MRT model is monotone and submodular.

Lemma 3.1. *Influence spread $\rho(\mathcal{S})$ for the MRT model satisfies (a) monotonicity: for any $\mathcal{S}^A \subseteq \mathcal{S}^B \subseteq \mathcal{V}$, $\rho(\mathcal{S}^A) \leq \rho(\mathcal{S}^B)$; and (b) submodularity: for any $\mathcal{S}^A \subseteq \mathcal{S}^B \subseteq \mathcal{V}$ and any pair $(v, t) \in \mathcal{V} \setminus \mathcal{S}^B$, $\rho(\mathcal{S}^A \cup \{(v, t)\}) - \rho(\mathcal{S}^A) \geq \rho(\mathcal{S}^B \cup \{(v, t)\}) - \rho(\mathcal{S}^B)$.*

Proof. The proof of monotonicity is straightforward, so we next consider submodularity. According to the definition of the influence spread, we have

$$\begin{aligned}
&\rho(\mathcal{S}^A \cup \{(v,t)\}) - \rho(\mathcal{S}^A) \geq \rho(\mathcal{S}^B \cup \{(v,t)\}) - \rho(\mathcal{S}^B) \\
\Leftrightarrow &\mathbb{E}\left[\left|\bigcup_{i=1}^{T} \Gamma(L_i, S_i^A) \setminus \Gamma(L_t, S_t^A) \cup \Gamma(L_t, S_t^A \cup \{v\})\right|\right] \\
&- \mathbb{E}\left[\left|\bigcup_{i=1}^{T} \Gamma(L_i, S_i^A)\right|\right] \\
\Leftrightarrow &\mathbb{E}\left[\left|\bigcup_{i=1}^{T} \Gamma(L_i, S_i^A) \cup \Gamma(L_t, S_t^A \cup \{v\})\right|\right] - \mathbb{E}\left[\left|\bigcup_{i=1}^{T} \Gamma(L_i, S_i^A)\right|\right] \\
\geq &\mathbb{E}\left[\left|\bigcup_{i=1}^{T} \Gamma(L_i, S_i^B) \setminus \Gamma(L_t, S_t^B) \cup \Gamma(L_{t+1}, S_t^A \cup \{v\})\right|\right] \\
&- \mathbb{E}\left[\left|\bigcup_{i=1}^{T} \Gamma(L_i, S_i^B)\right|\right] \\
\Leftrightarrow &\mathbb{E}\left[\left|\bigcup_{i=1}^{T} \Gamma(L_i, S_i^B) \cup \Gamma(L_t, S_t^B \cup \{v\})\right|\right] - \mathbb{E}\left[\left|\bigcup_{i=1}^{T} \Gamma(L_i, S_i^A)\right|\right].
\end{aligned}$$

Then it is sufficient to show that

$$\begin{aligned}
&\bigcup_{i=1}^{T} \Gamma(L_i, S_i^A) \cup \Gamma(L_t, S_t^A \cup \{v\}) \setminus \bigcup_{i=1}^{T} \Gamma(L_i, S_i^A) \\
&\supseteq \bigcup_{i=1}^{T} \Gamma(L_i, S_i^B) \cup \Gamma(L_t, S_t^B \cup \{v\}) \setminus \bigcup_{i=1}^{T} \Gamma(L_i, S_i^B). \qquad (1)
\end{aligned}$$

Algorithm 1: CR-Greedy: Cross-Round Greedy Algorithm

Input: Graph $G = (V, E)$, integers T, k and R, triggering set distributions.
Output: $\mathcal{S}^o$.

$\mathcal{S}^o \leftarrow \emptyset$; $C \leftarrow \mathcal{V}$;
$c_1, c_2, \ldots, c_t \leftarrow 0$; *// node counter for each round*
for $i = 1, 2, \ldots, kT$ **do**
  for all $(v, t) \in C \setminus \mathcal{S}^o$, estimate $\rho(\mathcal{S}^o \cup \{(v, t)\})$ by simulating the diffusion process R times;
  $(v_i, t_i) \leftarrow \text{argmax}_{(v,t)\in C\setminus\mathcal{S}^o} \hat{\rho}(\mathcal{S}^o \cup \{(v, t)\})$;
  $\mathcal{S}^o \leftarrow \mathcal{S}^o \cup \{(v_i, t_i)\}$; $c_{t_i} \leftarrow c_{t_i} + 1$;
  if $c_{t_i} \geq k$ **then** *// budget for round t_i exhausts*
    $C \leftarrow C \setminus \mathcal{V}_{t_i}$;
return $\mathcal{S}^o$.

For a node $u \in \bigcup_{i=1}^{T} \Gamma(L_i, S_i^B) \cup \Gamma(L_t, S_t^B \cup \{v\}) \setminus \bigcup_{i=1}^{T} \Gamma(L_i, S_i^B)$, u is reachable from $S_t^B \cup \{v\}$ in L_t, but not reachable from S_i^B in L_i for any $i \in [T]$. Thus u is also not reachable from S_i^A in L_i for any $i \in [t]$. Therefore, we conclude that $u \in \bigcup_{i=1}^{T} \Gamma(L_i, S_i^A) \cup \Gamma(L_t, S_t^A \cup \{v\}) \setminus \bigcup_{i=1}^{T} \Gamma(L_i, S_i^A)$. Thus the submodularity holds. □

The monotonicity and submodularity above are the theoretical basis of designing and analyzing greedy algorithms for the non-adaptive MRIM. In the following sections, we will consider two different settings separately for seed node selection: within-round and cross-round. For the within-round setting, one needs to determine the seed sets round by round, while for the cross-round setting, one is allowed to select nodes crossing rounds.

3.1 Cross-Round Setting

We design a greedy algorithm for the non-adaptive MRIM under cross-round setting, named CR-Greedy (Algorithm 1). The idea of CR-Greedy is that at every greedy step, it searches all (v, t) in the candidate space C and picks the one having the maximum marginal influence spread without replacement. If the budget for some round t exhausts, then the remaining nodes of $\mathcal{V}_t$ are removed from C. Note that as C contains nodes assigning to different rounds, CR-Greedy selects nodes crossing rounds.

Given a set U which is partitioned into disjoint sets $U_1, \ldots, U_n$ and $\mathcal{I} = \{X \subseteq U : |X \cap U_i| \leq k_i, \forall i \in [n]\}$, $(U, \mathcal{I})$ is called a *partition matroid*. Thus, the node space $\mathcal{V}$ with the constraint of MRIM, namely $(\mathcal{V}, \{\mathcal{S} : |S_t| \leq k, \forall t \in [T]\})$, is a partition matroid. This indicates that MRIM under cross-round setting is an instance of submodular maximization under partition matroid, and thus the performance of CR-Greedy has the following guarantee [13].

Theorem 3.2. *Let $\mathcal{S}^*$ be the optimal solution of the non-adaptive MRIM under cross-round setting. For every $\varepsilon > 0$ and $\ell > 0$, with probability at least $1 - \frac{1}{n^\ell}$, the output $\mathcal{S}^o$ of* CR-Greedy *satisfies*

$$\rho(\mathcal{S}^o) \geq \left(\frac{1}{2} - \varepsilon\right) \rho(\mathcal{S}^*),$$

if CR-Greedy *uses $R = \lceil 31k^2T^2n\log(2kn^{\ell+1})/\varepsilon^2 \rceil$ as input. In this case, the total running time is $O(k^3\ell T^4 n^2 m \log(n)/\varepsilon^2)$, assuming each simulation finishes in $O(m)$ time.*

Algorithm 2: WR-Greedy: Within-Round Greedy Algorithm

Input: Graph $G = (V, E)$, integers T, k and R, triggering set distributions.
Output: $\mathcal{S}^o$.

$\mathcal{S}^o \leftarrow \emptyset$;
for $t = 1, 2, \ldots, T$ **do**
  for $i = 1, 2, \ldots, k$ **do**
    for all $(v, t) \in \mathcal{V}_t \setminus \mathcal{S}^o$, estimate $\rho(\mathcal{S}^o \cup \{(v, t)\})$ by simulating the diffusion process R times;
    $(u, t) \leftarrow \text{argmax}_{(v,t)\in\mathcal{V}_t\setminus\mathcal{S}^o} \hat{\rho}(\mathcal{S}^o \cup \{(v, t)\})$;
    $\mathcal{S}^o \leftarrow \mathcal{S}^o \cup \{(u, t)\}$;
return $\mathcal{S}^o$.

3.2 Within-Round Setting

We give the second greedy algorithm (Algorithm 2) for the within-round setting, denoted as WR-Greedy. The idea of WR-Greedy is that seed nodes are selected round by round. More specifically, we greedily select seed nodes for the first round, and only after we selected k seed nodes for the first round, then we greedily select seed nodes for the next round, and so on. The immediate advantage of WR-Greedy over CR-Greedy is that in each greedy step the WR-Greedy only searches candidates (v, t) with t being the current round number, while CR-Greedy needs to search (v, t) for all rounds. This would give at least a factor of T saving on the running time of WR-Greedy. However, as we will show below, it pays a price of a slightly lower approximation ratio.

To analyze WR-Greedy, we utilize the result of Lemma 3.1 in a different way. First, when we fix the seed sets in round $1, \ldots, t-1$, and only vary the seed sets in round t, the influence spread certainly still satisfies the monotonicity and submodularity. Therefore, within round t, the seed set $\mathcal{S}_t^o$ selected by WR-Greedy for round t is a $(1 - 1/e - \varepsilon)$-approximate solution when the previous $t-1$ seed sets are fixed. Second, we could view seed set $\mathcal{S}_t$ of each round t as a unit, and when adding it to the previous units, it would also satisfy the monotonicity and submodularity by Lemma 3.1. Namely, $\rho(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_t) \leq \rho(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_{t'})$ for all $t < t'$, and $\rho(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_t \cup \mathcal{S}_{t''}) - \rho(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_t) \leq \rho(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_{t'} \cup \mathcal{S}_{t''}) - \rho(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_{t'})$, for all $t < t' < t''$. This means that WR-Greedy can be viewed as greedily selecting seed set units $\mathcal{S}_t$ round by round with the monotonicity and submodularity, while within each round, it can not find the optimal $\mathcal{S}_t$ but instead by a $(1 - 1/e - \varepsilon)$-approximate solution. Together, by the result in [26], we can show that WR-Greedy achieves an approximation factor of $1 - e^{-(1-1/e)} - \varepsilon$, as summarized in the following theorem.

Theorem 3.3. *Let $\mathcal{S}^*$ be the optimal solution of the non-adaptive MRIM under within-round setting. For every $\varepsilon > 0$ and $\ell > 0$, with probability at least $1 - \frac{1}{n^\ell}$, the output $\mathcal{S}^o$ of* WR-Greedy *satisfies*

$$\rho(\mathcal{S}^o) \geq \left(1 - e^{-(1-\frac{1}{e})} - \varepsilon\right) \rho(\mathcal{S}^*),$$

if WR-Greedy *uses $R = \lceil 31k^2n\log(2kn^{\ell+1}T)/\varepsilon^2 \rceil$ as input. In this case, the total running time is $O(k^3\ell T n^2 m \log(nT)/\varepsilon^2)$, assuming each simulation finishes in $O(m)$ time.*

Proof (Sketch). The proof follows the same structure as the proof of Theorem 3.7 in [7], but it needs to accommodate the new double greedy algorithm structure and the double submodular property. let $\varepsilon_0 = e^{(1-1/e)}\varepsilon/2$. From the inner-submoduarlity property of the MRT model and the proof of Theorem 3.7 in [7] based on the Chernoff bound, we can conclude that when $R \geq \lceil 27k^2 n \log(2kn^{\ell+1}T)/\varepsilon_0^2 \rceil$, for each round $t \in [T]$, with probability at least $1 - \frac{1}{n^\ell T}$, the seed set $\mathcal{S}_t^o$ found by WR-Greedy is a $(1 - 1/e - \varepsilon_0)$ approximation of the optimal solution for round i maximizing the marginal gain of $\rho(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_{t-1} \cup \mathcal{S}) - \rho(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_{t-1})$. Using the union bound, we know that with probability at least $1 - \frac{1}{n^\ell}$, for all $t \in [T]$ seed set S_t^o is an $(1 - 1/e - \varepsilon_0)$ approximation of the optimal solution for round t.

Now at each greedy step, if the new item found is not the one giving the best marginal contribution but an α approximation of the optimal marginal solution, an easy extension of [26], already reported in [15], show that the greedy algorithm can give a $1 - e^{-\alpha}$ approximate solution to the sumodular maximization problem. In our case, consider the outer level when treating each subset S_i as an item, Lemma 3.1 shows that ρ is also submodular in this case (the outer-submodularity), and we just argued in the previous paragraph that in each round, the selected $\mathcal{S}_t^o$ is a $(1-1/e-\varepsilon_0)$ approximation. Therefore, the final output $\mathcal{S}^o$ satisfies

$$\rho(\mathcal{S}^o) \geq (1 - e^{-(1-1/e-\varepsilon_0)})\rho(\mathcal{S}^*).$$

Because $\varepsilon_0 = e^{(1-1/e)}\varepsilon/2$, it is easy to verify that $1 - e^{-(1-1/e-\varepsilon_0)} \geq 1 - e^{-(1-1/e)} - \varepsilon$ in this case, and it is sufficient to have $R = \lceil 31k^2 n \log(2kn^{\ell+1}T)/\varepsilon^2 \rceil$. Finally, the total running time is simply $O(TknRm) = O(k^3 \ell T n^2 m \log(nT)/\varepsilon^2)$. □

Compared with Theorem 3.2, the approximation ratio drops from $1/2 - \varepsilon$ to $0.46 - \varepsilon$, but the running time improves by a factor of T^3. One factor of T is because each greedy step of CR-Greedy needs to search a space T times larger than that of WR-Greedy, and the other factor of T^2 is because CR-Greedy needs more accurate Monte Carlo estimates for each evaluation of $\rho(\mathcal{S})$ to avoid deviation, again because it searches a larger space. This shows the trade-off between efficiency and approximation ratio, that CR-Greedy has a better performance guarantee while WR-Greedy is much more efficient.

4 ADAPTIVE MRIM

We now study the adaptive MRIM problem. Informally, at the beginning of each round, one need to determine the seed set for the current round based on the propagation results observed in previous rounds. The formal definition follows the framework and terminology provided in [14] and will be given in Section 4.1. We then argue about the adaptive submodularity property, propose the adaptive greedy policy and analyze its performance in Section 4.2.

4.1 Notations and Definition

We call (S_t, t) an *item*, where S_t is the seed set chosen in round t. Let $\mathcal{E}$ be the set of all the possible items. For each item (S_t, t), after the propagation, the nodes and edges participated in the propagation are observed as the feedback. Formally, the feedback is referred to as a *state*, which is the subgraph of live-edge graph L_t that can be reached by S_t. A *realization* is a function $\phi : \mathcal{E} \to O$ mapping every possible item (S_t, t) to a state, where O is the set of all possible states. Realization ϕ represents one possible propagation from a possible seed set in a round. Let $\phi(S_t, t)$ denote the state of (S_t, t) under realization ϕ. We use Φ to denote a random realization, and the randomness comes from random live-edge graphs $L_1, \ldots, L_T$. For the adaptive MRIM, in each round t, we pick an item (S_t, t), see its state $\Phi(S_t, t)$, pick the next item $(S_{t+1}, t+1)$, see its state, and so on. After each pick, previous observations can be represented as a *partial realization* ψ, a function from some subset of $\mathcal{E}$ to their states. For notational convenience, we represent ψ as a relation, so that $\psi \subseteq \mathcal{E} \times O$ equals $\{((S_t, t), o) \colon \psi(S_t, t) = o\}$. We define $\mathrm{dom}(\psi) = \{(S_t, t) \colon \exists o, ((S_t, t), o) \in \psi\}$ as the *domain* of ψ. A partial realization ψ is *consistent with* a realization ϕ if they are equal everywhere in the domain of ψ, denoted as $\phi \sim \psi$. If ψ and ψ' are both consistent with some ϕ, and $\mathrm{dom}(\psi) \subseteq \mathrm{dom}(\psi')$, ψ is a *subrealization* of ψ', also denoted as $\psi \subseteq \psi'$.

A *policy* π is an adaptive strategy for picking items based on partial realizations in $\mathcal{E}$. In each round, π will pick the next set of seeds $\pi(\psi)$ based on partial realization ψ so far. If partial realization ψ is not in the domain of π, the policy stops picking items. We use $\mathrm{dom}(\pi)$ to denote the domain of π. Technically, we require that $\mathrm{dom}(\pi)$ be closed under subrealizations: If $\psi' \in \mathrm{dom}(\pi)$ and ψ is a subrealization of ψ' then $\psi \in \mathrm{dom}(\pi)$. We use the notation $E(\pi, \phi)$ to refer to the set of items selected by π under realization ϕ. The set of items in $E(\pi, \phi)$ is always in the form $\{(S_1, 1), \ldots, (S_t, t)\}$, so sometimes we also refer to it as sequence of seed sets.

We wish to maximize, subject to some constraints, a utility function $f \colon 2^{\mathcal{E}} \times O^{\mathcal{E}} \to \mathbb{R}_{\geq 0}$ that depends on the picked items and the states of them. In the adaptive MRIM, $f(\{(S_1, 1), \ldots, (S_t, t)\}, \phi)$ is the total number of active nodes by round t from the respective seed sets, i.e., $|\bigcup_{i=1}^t \Gamma(L_i^\phi, S_i)|$ where L_t^ϕ is the live-edge graph of round t. Based on the above notations, the expected utility of a policy π is $f_{\mathrm{avg}}(\pi) = \mathbb{E}_\Phi[f(E(\pi, \Phi), \Phi)]$ where the expectation is taken over the randomness of Φ. Namely, $f_{\mathrm{avg}}(\pi)$ is the expected number of active nodes under policy π. Let $\Pi_{T,k}$ be the set of all policies that select seed sets in at most T rounds and each seed set has at most k nodes. The goal of the adaptive MRIM is to find the best policy π such that: $\pi^* = \mathrm{argmax}_{\pi \in \Pi_{T,k}} f_{\mathrm{avg}}(\pi)$.

4.2 Adaptive Submodularity and Greedy Policy

Given a partial realization ψ of $t-1$ rounds with $\mathrm{dom}(\psi) = \{(S_1, 1), (S_2, 2), \ldots, (S_{t-1}, t-1)\}$, and the seed set S_t for round t, the *conditional expected marginal benefit* of item (S_t, t) conditioned on having observed ψ is defined as

$$\Delta((S_t, t)|\psi) = \mathbb{E}_\Phi\left[f(\mathrm{dom}(\psi) \cup \{(S_t, t)\}, \Phi) - f(\mathrm{dom}(\psi), \Phi) \mid \Phi \sim \psi\right].$$

The *conditional expected marginal gain* of a policy π is defined as

$$\Delta(\pi|\psi) = \mathbb{E}\left[f(\mathrm{dom}(\psi) \cup E(\pi, \Phi), \Phi) - f(\mathrm{dom}(\psi), \Phi) \mid \Phi \sim \psi\right].$$

The adaptive MRIM satisfies the adaptive monotonicity and submodularity shown as below. The proofs require a careful analysis of the partial realization in the MRT model and is given in [29].

Lemma 4.1. *[Adaptive Monotonicity] For all $t > 0$, for all partial realization ψ with $t-1$ rounds and $\Pr[\Phi \sim \psi] > 0$, and for all item (S_t, t), we have:*

$$\Delta((S_t, t)|\psi) \geq 0.$$

Algorithm 3: AdaGreedy: Adaptive Greedy for Round t

Input: Graph $G = (V, E)$, integers T, k and R, triggering set distributions, active node set A_{t-1} by round $t-1$.
Output: Seed set S_t, updated active nodes A_t

$S_t \leftarrow$ MC-Greedy(G, A_{t-1}, k, R); // *Monte Carlo Greedy*
Observe the propagation of S_t, update activated nodes A_t;
return $(S_t, t), A_t$.

Lemma 4.2. *[Adaptive Submodularity] For all $t > 0$, for all partial realization ψ with $i-1$ rounds and partial realization ψ' such that ψ' is a subrealization of ψ, i.e., $\psi' \subseteq \psi$, and for all item (S_i, i), we have*

$$\Delta((S_t, t)|\psi') \geq \Delta((S_t, t)|\psi).$$

Following the framework of [14], adaptive monotonicity and adaptive submodularity enable an adaptive greedy policy with a constant approximation of the optimal adaptive policy. AdaGreedy (Algorithm 3) is the greedy adaptive policy for MRIM. Note that adaptive algorithms operate at per round base — it takes feedback from previous rounds and selects the item for the current round, and then obtain new feedback. Thus we present AdaGreedy for a generic round t. Besides the problem input such as the graph G, the triggering model parameters, seed set budget k, and simulation number R, AdaGreedy takes the set of already activated nodes A_{t-1} as the feedback from the previous rounds, and aims at finding the seed set S_t of size k to maximize the expected marginal gain $\Delta((S_t, t)|\psi)$, which is the expected number of newly activated nodes in round t. However, this problem is NP-hard, so we use a Monte Carlo greedy approximation MC-Greedy algorithm to find an approximate solution. MC-Greedy greedily finds the seed with the maximum estimated marginal influence spread until k seeds being selected, where the marginal influence spread of adding an unselected seed is estimated by simulating the propagation R times. In AdaGreedy, MC-Greedy won't count the influence of a node if it has been activated in previous rounds. The rationale is that maximizing the expected marginal gain $\Delta((S_t, t)|\psi)$ is equivalent to the weighted influence maximization task in which we treat nodes in A_{t-1} with weight 0 and other nodes with weight 1, and we maximize the expected total weight of the influenced nodes. By [25], we know that the weighted version is also monotone and submodular, so we could use a greedy algorithm to obtain a seed set S_t as a $(1 - 1/e - \varepsilon)$ approximation of the best seed set for round t. We could use R Monte Carlo simulations to estimate the weighted influence spread, and R is determined by the desired approximation accuracy ε. Once the seed set S_t is selected for round t, the actual propagation from S_t will be observed, and the active node set A_t will be updated as the feedback for the next round.

The following theorem summarizes the correctness and the time complexity of AdaGreedy.

Theorem 4.3. *Let π^{ag} represents the policy corresponding to the* AdaGreedy *algorithm. For any $\varepsilon > 0$ and $\ell > 0$, if we use $R = \lceil 31k^2 n \log(2kn^{\ell+1}T)/\varepsilon^2 \rceil$ simulations for each influence spread estimation, then with probability at least $1 - \frac{1}{n^\ell}$,*

$$f_{\mathrm{avg}}(\pi^{\mathrm{ag}}) \geq \left(1 - e^{-(1-\frac{1}{e})} - \varepsilon\right) f_{\mathrm{avg}}(\pi^*).$$

Algorithm 4: CR-NS: Cross-Round Node Selection

Input: Multi round RR vector sets $\mathcal{M}$, T, k
Output: seed sets $\mathcal{S}^o$

Build count array: $c[(u,t)] = \sum_{(u,t)\in\mathcal{M}} |(u,t)|$, $\forall (u,t) \in \mathcal{V}$;
Build RR set link: $RR[(u,t)]$, $\forall (u,t) \in \mathcal{V}$;
For all $\mathcal{R} \in \mathcal{M}$, $covered[\mathcal{R}] = false$;
$\mathcal{S}^o \leftarrow \emptyset$; $C \leftarrow \mathcal{V}$; $c_1, c_2, \ldots, c_T \leftarrow 0$;
for $i = 1$ *to* Tk **do**
  $(u,t) \leftarrow \mathrm{argmax}_{(u',t')\in C\setminus\mathcal{S}^o} c[(u',t')]$;
  $\mathcal{S}^o \leftarrow \mathcal{S}^o \cup \{(u,t)\}$; $c_t \leftarrow c_t + 1$;
  if $c_t == k$ **then**
    $C = C \setminus \{(v,t) \mid v \in V\}$
  for *all* $\mathcal{R} \in RR[(u,t)] \wedge covered[\mathcal{R}] = false$ **do**
    $covered[\mathcal{R}] = true$;
    for all $(u',t') \in \mathcal{R} \wedge (u',t') \neq (u,t)$ to do
    $c[(u',t')] = c[(u',t')] - 1$;
return $\mathcal{S}^o$.

In this case, the total running time for T-round AdaGreedy *is $O(k^3 \ell T n^2 m \log(nT)/\varepsilon^2)$.*

5 SCALABLE IMPLEMENTATIONS

In this section, we aim to speed up CR-Greedy, WR-Greedy and AdaGreedy by the reverse influence sampling [4, 30].

A *Reverse-Reachable (RR) set R_v rooted at node $v \in V$* is the set of nodes which are reached by reverse simulating a propagation from v in the triggering model. Equivalently, R_v is the set of nodes in a random live-edge graph which can reach v. We use root(R_v) to denote its root v. We define a *(random) RR set R* is a RR set rooted at a node picked uniformly at random from V, then for any seed set $S \subseteq V$, its influence spread

$$\sigma(S) = n \cdot \mathbb{E}[\mathbb{I}\{S \cap R \neq \emptyset\}], \tag{2}$$

where $n = |V|$, $\mathbb{I}\{\}$ is the indicator function, and the expectation is taken over the randomness of R: randomness of root node and randomness of live-edge graph. The property implies that we can accurately estimate the influence spread of any possible seed set S by sampling enough RR sets. More importantly, by Eq. (2) the optimal seed set can be found by seeking the optimal set of nodes that intersect with (a.k.a. cover) the most number of RR sets, which is a max-cover problem. Therefore, a series of near-linear-time algorithms are developed [4, 30] based on the above observation.All RR-set algorithms have the same structure of two phases. In Phase 1, the number of RR sets needed is estimated, and in Phase 2, these RR sets are generated and greedy algorithm is used on these RR sets to find the k nodes that cover the most number of RR sets. All algorithms have the same Phase 2, but Phase 1 is being improved from one to another so that less and less RR sets are needed. Our algorithms are based on IMM proposed in [30].

5.1 Non-Adaptive IMMs

For the non-adaptive MRIM, we define the *multi-round reverse-reachable (RR) set $\mathcal{R}_v$ rooted at node v* for the MRT model as $\mathcal{R}_v :=$

Algorithm 5: CR-NAIMM: Non-adaptive IMM Algorithm for Cross-Round

Input: Graph $G = (V, E)$, round number T, budget k, accuracy parameters (ε, ℓ), triggering set distributions
Output: seed set $\mathcal{S}$

// Phase 1: Estimating the number of multi-round RR sets needed, θ

1 $\ell \leftarrow \ell + \ln 2/\ln n$; $\mathcal{M} \leftarrow \emptyset$; $LB \leftarrow 1$; $\varepsilon' \leftarrow \sqrt{2}\varepsilon$;
2 $\alpha \leftarrow \sqrt{\ell \ln n + \ln 2}$; $\beta \leftarrow \sqrt{(1-1/2)\cdot(T\ln\binom{n}{k} + \alpha^2)}$;
3 $\lambda' \leftarrow [(2+\frac{2}{3}\varepsilon')\cdot(T\ln\binom{n}{k} + \ell\cdot\ln n + \ln\log_2 n)\cdot n]/\varepsilon'^2$;
4 $\lambda^* \leftarrow 2nT\cdot((1-1/e)\cdot\alpha+\beta)^2\cdot\varepsilon^{-2}$;
5 **for** $i = 1$ *to* $\log_2(n-1)$ **do**
6 $\quad x \leftarrow n/2^i$;
7 $\quad \theta_i \leftarrow \lambda'/x_i$;
8 $\quad$ **while** $|\mathcal{M}| < \theta_i$ **do**
9 $\quad\quad$ Select a node u from V uniformly at random;
10 $\quad\quad$ Generate RR-vector $\mathcal{R}$ from u, and insert it into $\mathcal{M}$;
11 $\quad \mathcal{S}_i \leftarrow$ CR-NS$(\mathcal{M}, T, k)$;
12 $\quad$ **if** $n\cdot F_{\mathcal{M}}(\mathcal{S}_i) \geq (1+\varepsilon')\cdot x$ **then**
13 $\quad\quad LB \leftarrow n\cdot F_{\mathcal{M}}(\mathcal{S}_i)/(1+\varepsilon')$;
14 $\quad\quad$ break;
15 $\theta \leftarrow \lambda^*/LB$;
16 **while** $|\mathcal{M}| \leq \theta$ **do**
17 $\quad$ Select a node u from V uniformly at random;
18 $\quad$ Generate $\mathcal{R}$ for u, and insert it into $\mathcal{M}$;

// Phase 2: Generate θ RR-vector sets and select seed nodes

19 $\mathcal{S} \leftarrow$ CR-NS$(\mathcal{M}, T, k)$;
20 **return** $\mathcal{S}$.

$\bigcup_{t=1}^{T}\mathcal{R}_{v,t}$ where $\mathcal{R}_{v,t}$ denotes a RR set rooted at v of round t in pair notation. $\mathcal{R}_v$ is generated by independently reverse simulating the propagation T rounds from v and then aggregating them together. Let root$(\mathcal{R}_v) := v$. A (random) multi-round RR set $\mathcal{R}$ is a multi-round RR set rooted at a node picked uniformly at random from V. We use $\mathcal{M}$ to denote the set of $\mathcal{R}_v$. We are now ready to explain the cross-round and within-round non-adaptive IMM.

5.1.1 Cross-Round Non-Adaptive IMM. In cross-round setting, if $\mathcal{R}$ is a random multi-round RR set, then for any seed set $\mathcal{S}$, its influence spread satisfies the following lemma.

Lemma 5.1. *For any node-round pair seed set $\mathcal{S}$,*

$$\rho(\mathcal{S}) = n\cdot\mathbb{E}[\mathbb{I}\{\mathcal{S}\cap\mathcal{R}\neq\emptyset\}],$$

where the expectation is taken over the randomness of $\mathcal{R}$.

The Lemma 5.1 implies we can sample enough multi-round RR sets to accurately estimate the influence spread of $\mathcal{S}$.

CR-NAIMM is very similar to standard IMM and only has a few differences include several points. First, CR-NAIMM generates multi-round RR sets $\mathcal{R}$ from roots in V (lines 9 and 17). Second, we need to adjust ℓ to be $\ell + \log(2)/\log n$, and ε. This is to guarantee that in each round we have probability at least $1 - 1/(n^\ell T)$ to have $\mathcal{S}_t$ as a $(\frac{1}{2}-\varepsilon)$ approximation, so that the result for the total T rounds would come out correctly as stated in the following theorem. Third, we use new $F_{\mathcal{M}}(\cdot)$ denotes the fraction of multi-round RR sets in $\mathcal{R}$ that are covered by a node set $\mathcal{S}$ in algorithm CR-NAIMM (lines 12 and 13). Forth, CR-NS returns Tk seeds from the total T rounds (line 5). Last, if the budget for some round t exhausts, then the remaining nodes of $\mathcal{V}_t$ are removed from C in CR-NS (line 9).

By an analysis similar to that of the IMM algorithm [30], we can show that our CR-NAIMM achieves $1/2-\varepsilon$ approximation with expected running time $O(T^2(k+\ell)(m+n)\log(n)/\varepsilon^2)$.

Theorem 5.2. *Let $\mathcal{S}^*$ be the optimal solution of the non-adaptive MRIM. For every $\varepsilon > 0$ and $\ell > 0$, with probability at least $1 - \frac{1}{n^\ell}$, the output $\mathcal{S}^o$ of the cross-round algorithm* CR-NAIMM *satisfies*

$$\rho(\mathcal{S}^o) \geq \left(\frac{1}{2}-\varepsilon\right)\rho(\mathcal{S}^*),$$

In this case, the total running time for T-round CR-NAIMM *is $O(T^2(k+\ell)(n+m)\log(n)/\varepsilon^2)$.*

5.1.2 Within-Round Non-Adaptive IMM. In the within-round setting, the idea is to use the IMM algorithm in each round to select k seeds. However, seeds selected in earlier rounds may already influence some nodes, so when we select roots for a later round t and generate RR sets for round t, the roots should not be selected uniformly at random. Instead, we want to utilize the idea derived from the following lemma. Let $\mathcal{S}^{t-1} := \bigcup_{t'=1}^{t-1}\mathcal{S}_{t'}$ be the set of seed pairs in first $t-1$ rounds, and let $\mathcal{S}_t$ be a set of seed pairs in round t. Similarly, let $\mathcal{R}_v^{t-1} := \bigcup_{t'=1}^{t-1}\mathcal{R}_{v,t'}$ be the set of RR sets (in pairs) in first $t-1$ rounds, $\mathcal{R}_t$ be the RR set (in pairs) for round t. The marginal influence spread of $\mathcal{S}_t$ in round t is

Lemma 5.3. *For any node-round pair seed set $\mathcal{S}$,*

$$\begin{aligned}&\rho(\mathcal{S}^{t-1}\cup\mathcal{S}_t) - \rho(\mathcal{S}^{t-1})\\ &= n\cdot\mathbb{E}[\mathbb{I}\{(\mathcal{S}^{t-1}\cap\mathcal{R}^{t-1}=\emptyset)\wedge(\mathcal{S}_t\cap\mathcal{R}_t\neq\emptyset)\}]\\ &= n\cdot\Pr\{\mathcal{S}^{t-1}\cap\mathcal{R}^{t-1}=\emptyset\}\cdot\mathbb{E}[\mathbb{I}\{\mathcal{S}_t\cap\mathcal{R}_t\neq\emptyset\}\mid\mathcal{S}^{t-1}\cap\mathcal{R}^{t-1}=\emptyset],\end{aligned}$$

where the expectation is taken over the randomness of $\mathcal{R}$.

The above lemma suggests that when we want to generate an RR set for round t, we should also generate RR sets for earlier rounds and check that if any of them is intersecting with the seed set in the same round, and if so, the RR set for round t is invalid and we need to regenerate an RR set again. By following this, the implementation is similar to the CR-NAIMM, and the only difference between them is that the within-round non-adaptive algorithm WR-NAIMM selects k seeds round by round. The resulting algorithm would have the approximation guarantee of $1 - e^{-(1-\frac{1}{e})} - \varepsilon$, but it does not have significant running time improvement over CR-NAIMM, since it wastes many RR set generations. We find a better heuristic to use the roots generated for the previous rounds. In particular, for each $t \geq 2$, after we finished selecting k seeds in round $t-1$, some RR sets are removed since they are covered by seeds selected. The remaining roots are exactly the ones whose RR sets do not intersect with seed sets in the first $t-1$ rounds. Therefore, their distribution is close to the distribution of the valid roots satisfying $\mathcal{S}^{t-1}\cap\mathcal{R}^{t-1}=\emptyset$ in Lemma 5.3. Hence, in round t, we sample roots using the remaining roots from round $t-1$. This gives a close estimate of the marginal influence spread. Due to the complicated stochastic dependency

of RR sets from round to round, the exact theoretical analysis of this improvement is beyond our reach, and thus we propose it as an efficient heuristic.

The resulting algorithm WR-NAIMM runs almost exactly like running a copy of the standard IMM for each round t. The only difference is that in the standard IMM, the root of an RR set is always sampled uniformly at random from all nodes, but in WR-NAIMM the root of an RR set in round t is sampled uniformly at random from the remaining roots left in round $t-1$. Moreover, for each round t, we set the per-round approximation error bound $\varepsilon_0 = e^{(1-1/e)}\varepsilon/2$ and replace ℓ with $\ell + \log(2T)/\log n$. This is consistent with the adaptive IMM setting in Section 5.2, and with the setting used in the proof of Theorem 3.3 for the within-round greedy algorithm. The details of WR-NAIMM are provided in [29].

5.2 Adaptive IMM

We use M to denote the set of R_v. In the adaptive setting, as already mentioned in Section 4.2, some nodes A_{t-1} are already activated, so they won't contribute to the influence spread in round t. Therefore, we are working on the weighted influence maximization problem, where nodes in A_{t-1} has weight 0 and nodes in $V \setminus A_{t-1}$ has weight 1. Let $\sigma^{-A_{t-1}}(S)$ be the weighted influence spread according to the above weight. Let $R^{-A_{t-1}}$ be a random RR set where the root v is selected from $V \setminus A_{t-1}$ uniformly at random, and then reverse simulate from v to get the RR set. The result is similar to Eq. (2):

Lemma 5.4. *For any seed set $S \subseteq V$,*

$$\sigma^{-A_{t-1}}(S) = (n - |A_{t-1}|) \cdot \mathbb{E}[\mathbb{I}\{S \cap R^{-A_{t-1}} \neq \emptyset\}], \tag{3}$$

where the expectation is taken over the randomness of $R^{-A_{t-1}}$.

Proof.

$$\begin{aligned}
&\mathbb{E}[\mathbb{I}\{S \cap R^{-A_{i-1}} \neq \emptyset\}] \\
&= \sum_{v \in V \setminus A_{i-1}} \Pr\{v = \text{root}(R^{-A_{i-1}})\} \cdot \\
&\qquad \mathbb{E}[\mathbb{I}\{S \cap R^{-A_{i-1}} \neq \emptyset\} \mid v = \text{root}(R^{-A_{i-1}})] \\
&= \frac{1}{n - |A_{i-1}|} \sum_{v \in V \setminus A_{i-1}} \mathbb{E}[\mathbb{I}\{S \cap R_v^{-A_{i-1}} \neq \emptyset\}] \\
&= \frac{1}{n - |A_{i-1}|} \sum_{v \in V \setminus A_{i-1}} \mathbb{E}[\mathbb{I}\{v \in \Gamma(L, S)\}] \qquad (4) \\
&= \frac{1}{n - |A_{i-1}|} \mathbb{E}\left[\sum_{v \in V \setminus A_{i-1}} \mathbb{I}\{v \in \Gamma(L, S)\}\right] \\
&= \frac{1}{n - |A_{i-1}|} \mathbb{E}[|\Gamma(L, S) \setminus A_{i-1}|] \\
&= \frac{1}{n - |A_{i-1}|} \sigma^{-A_{i-1}}(S),
\end{aligned}$$

where Eq. (4) is based on the equivalence between RR sets and live-edge graphs, and the expectation from this point on is taken over the random live-edge graphs L. □

Therefore, with Eq. (3), the same RR-set based algorithm can be used, and we only need to properly change the RR-set generation process and the estimation process. AdaIMM is based on the IMM algorithm in [30]. The main differences from the standard IMM include several points. First, whenever we generate new RR sets in round t, we only start from roots in $V \setminus A_{t-1}$ as explained by Lemma 5.4. Second, when we estimate the influence spread, we need to adjust it using $n_a = n - |A_{t-1}|$ again by Lemma 5.4. Third, we need to adjust ℓ to be $\ell + \log(2T)/\log n$, and ε to $\varepsilon_0 = e^{(1-1/e)}\varepsilon/2$. This is to guarantee that in each round we have probability at least $1 - 1/(n^\ell T)$ to have S_t as a $(1 - 1/e - \varepsilon_0)$ approximation, so that the result for the total T rounds would come out correctly as stated in the following theorem. The details are shown in [29].

Theorem 5.5. *Let π^{ai} represents the policy corresponding to the* AdaIMM *algorithm. For any $\varepsilon > 0$ and $\ell > 0$, with probability at least $1 - \frac{1}{n^\ell}$,*

$$f_{\text{avg}}(\pi^{\text{ai}}) \geq \left(1 - e^{-(1-\frac{1}{e})} - \varepsilon\right) f_{\text{avg}}(\pi^*).$$

In this case, the total running time for T-round AdaIMM *is $O(T(k + \ell)(n + m)\log(nT)/\varepsilon^2)$.*

Proof (Sketch). Let $\varepsilon_0 = e^{(1-1/e)}\varepsilon/2$, $\ell' = \ell + \log(2T)/\log n$. As explained already, one round AdaIMM is essentially the same as IMM with parameters ε_0 and ℓ'. Thus, following the result in [30], we know that for each round i, with probability at least $1 - 2/n^{\ell'} = 1 - 1/(n^\ell T)$, output S_i is a $(1 - 1/e - \varepsilon_0)$ approximation of the best seed set for this round. Then following the similar arguments as in Theorems 3.3 and 4.3, we know that across all T rounds, with probability at least $1 - 1/n^\ell$,

$$\begin{aligned}
f_{\text{avg}}(\pi^{\text{ai}}) &\geq \left(1 - e^{-(1-\frac{1}{e}-\varepsilon_0)}\right) f_{\text{avg}}(\pi^*) \\
&\geq \left(1 - e^{-(1-\frac{1}{e})} - \varepsilon\right) f_{\text{avg}}(\pi^*).
\end{aligned}$$

Thus, the theorem holds. □

Theorem 5.5 clearly shows that the AdaIMM algorithm is near linear time, and its theoretical time complexity bound is much better than the one in Theorem 4.3 for the AdaGreedy algorithm.

In practice, we can further improve the AdaIMM algorithm by incremental computation. In particular, RR sets generated in the previous $t-1$ rounds can be used for round t, and we only need to remove those rooted at nodes in A_{t-1}. Searching the lower bound LB can also be made faster by utilizing the x_{t-1} already obtained in the previous round. Our experiments would use such improvements.

6 EXPERIMENTAL EVALUATION

We conduct experiments on two real-world social networks to test the performance of our algorithms. We use the independent cascade model for the tests. The influence probabilities on the edges are learned from the real-world trace data in one dataset and synthetic in the other dataset, as explained below.

6.1 Data Description

Flixster. The Flixster dataset is a network of American social movie discovery service (www.flixster.com). To transform the dataset into a weighted graph, each user is represented by a node, and a directed edge from node u to v is formed if v rates one movie shortly after u does so on the same movie. The dataset is analyzed in [1], and the influence probability are learned by the topic-aware model. We

use the learning result of [1] in our experiment, which is a graph containing 29,357 nodes and 212,614 directed edges.

NetHEPT. The NetHEPT dataset [5] is extensively used in many influence maximization studies. It is an academic collaboration network from the "High Energy Physics Theory" section of arXiv from 1991 to 2003, where nodes represent the authors and each edge represents one paper co-authored by two nodes. There are 15,233 nodes and 58,891 undirected edges (including duplicated edges) in the NetHEPT dataset. We clean the dataset by removing those duplicated edges and obtain a directed graph $G = (V, E)$, $|V|$ = 15,233, $|E|$ = 62,774 (directed edges).

6.2 Result

We test all six algorithms proposed in the experiment, for the MRIM task of T rounds with k seeds in each round. We use $R = 10000$ for the Monte Carlo simulation of the influence spread of each candidate seed set for all non-adaptive algorithms. The lazy evaluation technique of [22] to optimize the greedy selection is applied to CR-Greedy, WR-Greedy and AdaGreedy. Besides the six proposed algorithms, we also propose two baseline non-adaptive algorithms (SG and SG-R) using the classical single-round algorithms directly for the multi-round influence maximization problem. SG simply selects Tk seed nodes using the single-round greedy algorithm, and then allocates the first k seeds as S_1 for the first round, second k seeds as S_2 for the second round, and so on. SG-R only selects k seeds greedily, and then reuse the same k seeds for all T rounds.

In the tests, we set $T = 5$ and $k = 10$ as the default, and we also test different combinations of T and k while keeping Tk to be the same, to see the effect of different degrees of adaptiveness.

6.2.1 Influence Spread Performance. We test the performance on the influence spread for all algorithms introduced in the above section. For non-adaptive algorithms, for each selected seed set sequence, we do 10,000 forward simulations and take the average to obtain its estimated influence spread. For adaptive algorithms, to obtain their expected influence spread over multiple real-world propagation simulations, we have to re-run the algorithm in each round after obtaining the feedback from the previous rounds. Thus, it would be too time consuming to also run 10000 adaptive simulations for the adaptive algorithms. In stead, for NetHEPT we use 150 simulations and for Flixster we use 100 simulations. To make fair comparisons, we include confidence intervals in the obtained influence spread results, so that the number of simulations used for the estimation is taken into the consideration.

Tables 1 and 2 show the influence spread results for NetHEPT and Flixster datasets. All five round results are shown, one for each column. Each row is for one algorithm, and the number in the first line of the row is the empirical average of the influence spread, and the line below is the 95% confidence interval. Parameter R records the number of simulations used to obtain the average spread.

Several observations can be made from these results. First, all six proposed algorithms in this paper performs significantly better than the baseline algorithms SG and SG-R. Besides the cross-round non-adaptive algorithms, the confidence intervals do not overlap starting from round 2. In terms of the empirical average, the improvement is obvious: at the end of the 5th round, for NetHEPT, MRIM algorithms is at least 8.8% better than SG, and 7.3% better than SG-R; for Flixster, MRIM algorithms is at least 12.8% better than SG, and 4.3% better than SG-R. Even if we use the ratio between the lower confidence bounds of MRIM algorithms vs. the upper confidence bounds of the baseline algorithms, the result is similar. SG-R performs better than SG, which implies that influential nodes are important in these datasets and it is preferred to re-select them. However, the improvement of MRIM algorithms over SG and SG-R are increasing over rounds, showing that adjusting to MRIM is increasingly important. It is reasonable to see that with more rounds, always sticking to the same seed sets or always changing seed sets would not perform well.

Table 1: The performance of influence spread on NetHEPT.

Method/Simulations	Round				
	1	2	3	4	5
SG	290.1	505.7	688.6	868.2	1027.3
(R = 10000)	[288.8, 291.4]	[504.0, 507.3]	[686.6, 690.4]	[866.2, 870.2]	[1025.2, 1029.4]
SG-R	289.5	516.3	714.0	884.9	1042.0
(R = 10000)	[288.2, 290.8]	[514.6, 518.0]	[712.0, 716.0]	[882.7, 887.1]	[1039.7, 1044.2]
E-WR-Greedy	290.7	528.9	738.8	930.2	1097.6.9
(R = 10000)	[289.4, 292.0]	[527.2, 530.6]	[736.9, 740.8]	[928.0, 932.3]	[1095.3, 1099.8]
WR-IMM	290.9	532.8	745.3	930.1	1093.1
(R = 10000)	[289.7, 292.3]	[531.1, 534.5]	[743.2, 747.3]	[928.0, 932.2]	[1090.8, 1095.3]
CR-Greedy	267.8	528.7	730.4	938.5	1121.3
(R = 10000)	[266.5, 269.1]	[527.2, 530.4]	[728.5, 732.4]	[933.7, 937.8]	[1119.0, 1123.5]
CR-IMM	283.0	517.4	721.9	931.6	1129.7
(R = 10000)	[281.7, 284.2]	[515.7, 519.2]	[720.0, 723.9]	[929.4, 933.7]	[1127.7, 1131.9]
AdaGreedy	288.3	533.4	758.1	960.1	1141.5
(R = 150)	[276.7, 299.7]	[519.4, 547.3]	[743.6, 772.7]	[943.9, 976.3]	[1123.7, 1160.0]
AdaIMM	**291.8**	**544.4**	**761.8**	**965.8**	**1146.3**
(R = 150)	[281.3, 302.4]	[531.6, 557.2]	[746.6, 776.9]	[949.7, 982.0]	[1129.1, 1163.5]

Table 2: The performance of influence Spread on Flixster.

Method/Simulations	Round				
	1	2	3	4	5
SG	558.8	936.2	1200.3	1437.9	1631.5
(R = 10000)	[557.3, 560.3]	[934.5, 937.9]	[1198.4, 1202.2]	[1435.9, 1439.9]	[1629.5, 1633.6]
SG-R	**559.8**	949.2	1262.6	1530.3	1764.9
(R = 10000)	[558.3, 561.3]	[947.4, 951.0]	[1260.6, 1264.5]	[1528.2, 1532.4]	[1762.7, 1767.0]
E-WR-Greedy	557.8	976.5	1304.2	1587.8	1840.0
(R = 10000)	[556,3 559.2]	[974.8, 978,3]	[1302.2, 1306.1]	[1585.8, 1580.8]	[1838.0, 1842.1]
WR-IMM	558.1	967.5	1306.9	1599.1	1836.4
(R = 10000)	[556.7, 559.6]	[965.7, 969.3]	[1306.9, 1308.9]	[1597.1, 1601.1]	[1834.3, 1838.5]
CR-Greedy	519.9	948.6	1295.7	1593.5	1863.8
(R = 10000)	[518.4, 521.5]	[946.7, 950.5]	[1293.7, 1297.7]	[1591.4, 1595.5]	[1861.7, 1865.9]
CR-IMM	521.7	935.8	1275.3	1585.9	1865.1
(R = 10000)	[521.7, 523.2]	[933.1, 937.0]	[1273.3, 1277.3]	[1583.8, 1588.0]	[1863.1, 1867.3]
AdaGreedy	557.8	977.8	1307.7	1605.2	1861.8
(R = 100)	[539.8, 580.5]	[956.2, 999.1]	[1291.1, 1324.3]	[1588.1, 1622.3]	[1845.3, 1878.3]
AdaMM	555.5	**977.9**	**1317.2**	**1613.2**	**1872.5**
(R = 100)	[542.3, 568.6]	[962.9, 993.0]	[1300.8, 1333.5]	[1594.2, 1632.1]	[1853.0, 1891.9]

Second, the adaptive algorithms perform better than the within-round non-adaptive algorithms in both dataset: the confidence interval does not overlap for all results in round 4 and 5 and most results in round 3. This conforms with our intuition that adaptive algorithms performs better. The improvement are not very significant in Tables 1 and 2, which means the strength of the adaptiveness has not be fully explored yet.

Third, the cross-round setting is more effective, but less efficient than within-round setting of non-adaptive algorithms, due to a larger search space. In practice, the cross-round setting only shows outstanding performances overall, but doesn't guarantee the good performance in every round. In fact, it always performs worst in round 1 comparing to the other algorithms.

Fourth, the cross-round non-adaptive algorithm performs as well as the adaptive algorithm on NetHept dataset. Adaptive algorithm

Table 3: Influence spread with different adaptive degree.

Num. of Rounds	1	2	5	10
Num. of Seeds	50	25	10	5
AdaIMM (R = 100)	883.0 [856.0, 910.1]	1040.3 [1022.6, 1058.1]	1141.0 [1119.3, 1162.6]	1204.7 [1178.2, 1231.3]

Table 4: Running time of the algorithms, in seconds.

	SG	SG-R	E-WR-Greedy	WR-IMM
NetHEPT (R = 5)	439.2 [407, 470.94]	87.8 [81.5, 94.2]	551.2 [527.9, 574.4]	1.97 [1.91, 2.03]
Flixster (R = 5)	4862.3 [4773.3, 4951.3]	972.5 [990.3,954.7]	2478.9 [2422.4, 2535.5]	3.16 [3.14, 3.18]

	CR-Greedy	CR-IMM	AdaGreedy	AdaIMM
NetHEPT (R = 5)	2105.6 [2036.2, 2175.0]	2.13 [2.05, 2.21]	465.4 [473.8, 457.0]	2.01 [1.93, 2.09]
Flixster (R = 5)	9587.6 [9145.3.10029.9]	3.61 [3.59, 3.63]	2305.5 [2161.0, 2450.0]	3.23 [3.16, 3.30]

requires the real-life spread between each round comparing to non-adaptive algorithms, but always performs the best in each round.

Fifth, we can see that AdaIMM achieves the same level of influence spread as AdaGreedy: the confidence intervals always overlap, which means AdaIMM performs well in practice.

6.2.2 Degree of Adaptiveness. We vary the parameters T and k whiling keeping Tk the same. With smaller k, it means each round we select a smaller number of seed sets, and we use more adaptive rounds. Therefore, small k and large T mean a high degree of adaptiveness. We test this using AdaIMM, since it is more efficient than AdaGreedy while providing the same level of influence spread.

Table 3 presents the result on this test, in which we vary (T, k) as $(1, 50)$, $(2, 25)$, $(5, 10)$ and $(10, 5)$. The influence spread significantly increases with the increase of the degree of adaptiveness, and the increase is quite significant. The 10-round 5-seed setup is 36.3% better than 1 round 50-seed setup on the empirical average. This shows that higher adaptive degree indeed improves the performance.

In summary, with same total budget, the higher number of total rounds with higher influence spread performance in general.

6.2.3 Running Time. Table 4 reports the running time of all methods on the two datasets, when running with $T = 5$ and $k = 10$. One conclusion is that all IMM algorithms are much more efficient that others, with two to three orders of magnitude faster than all other algorithms. Among greedy algorithms, SG-R runs faster because it selects only 10 seeds once. SG is the slowest, much slower than E-WR-Greedy and AdaGreedy because it selects 50 different seeds. When it selects more seeds, their marginal influence spread doesn't differ from one another much and thus the lazy-evaluation optimization is not as effective as selecting the first 10 seeds.

After combining the influence spread and running time performance, our conclusion is that (a) algorithms designed for the MRIM task is better, (b) the cross-round setting is more effective, but less efficient than the within-round setting of non-adaptive algorithms, and (c) AdaIMM is clearly the best for adaptive MRIM task.

7 ACKNOWLEDGE

This work is supported in part by NSF through grants IIS-1526499, IIS-1763325, and CNS-1626432, and NSFC 61672313, 61433014.

REFERENCES

[1] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. 2012. Topic-aware social influence propagation models. In *ICDM'12*. IEEE.
[2] Jonah Berger. 2014. Word of mouth and interpersonal communication: A review and directions for future research. *Journal of Consumer Psychology* (2014).
[3] Smriti Bhagat, Amit Goyal, and Laks V. S. Lakshmanan. 2012. Maximizing product adoption in social networks. In *WSDM*. ACM.
[4] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing social influence in nearly optimal time. In *ACM-SIAM (SODA '14)*.
[5] Ning Chen. 2008. On the approximability of influence in social networks. In *SODA '08*. Society for Industrial and Applied Mathematics.
[6] Wei Chen, Alex Collins, Rachel Cummings, Te Ke, Zhenming Liu, David Rincon, Xiaorui Sun, Yajun Wang, Wei Wei, and Yifei Yuan. 2011. Influence maximization in social networks when negative opinions may emerge and propagate. In *SDM*.
[7] Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. 2013. *Information and Influence Propagation in Social Networks*. Morgan & Claypool Publishers.
[8] Wei Chen, Tian Lin, Zihan Tan, Mingfei Zhao, and Xuren Zhou. 2016. Robust Influence Maximization. In *KDD*.
[9] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *KDD*.
[10] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. 2016. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research* (2016).
[11] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In *ICDM*.
[12] Pedro Domingos and Matthew Richardson. 2001. Mining the network value of customers. In *KDD*.
[13] Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. 1978. An analysis of approximations for maximizing submodular set functions II. In *Polyhedral combinatorics*. Springer.
[14] Daniel Golovin and Andreas Krause. 2011. Adaptive Submodularity:Theory and Applications in Active Learning and Stochastic Optimization. *JAIR* (2011).
[15] P. R. Goundan and A. S. Schulz. 2007. *Revisiting the greedy approach to submodular set function maximization*. Technical Report. MIT.
[16] Lifang He, Chun-Ta Lu, Jiaqi Ma, Jianping Cao, Linlin Shen, and Philip S Yu. 2016. Joint community and structural hole spanner detection via harmonic modularity. In *SIGKDD*. ACM.
[17] X. He and D. Kempe. 2016. Robust Influence Maximization. In *KDD*.
[18] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. 2012. Influence Blocking Maximization in Social Networks under the Competitive Linear Threshold Model.
[19] Kyomin Jung, Wooram Heo, and Wei Chen. 2012. IRIE: Scalable and Robust Influence Maximization in Social Networks. In *ICDM*.
[20] David Kempe, Jon M. Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *KDD*.
[21] Siyu Lei, Silviu Maniu, Luyi Mo, Reynold Cheng, and Pierre Senellart. 2015. Online Influence Maximization. In *KDD*.
[22] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne M. VanBriesen, and Natalie S. Glance. 2007. Cost-effective outbreak detection in networks. In *KDD*.
[23] Su-Chen Lin, Shou-De Lin, and Ming-Syan Chen. 2015. A learning-based framework to handle multi-round multi-party influence maximization on social networks. In *KDD*. ACM.
[24] Wei Lu, Wei Chen, and Laks VS Lakshmanan. 2015. From competition to complementarity: comparative influence diffusion and maximization. *PVLDB* (2015).
[25] Elchanan Mossel and Sebastien Roch. 2007. On the submodularity of influence in social networks. In *STOC '07*.
[26] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. 1978. *An analysis of the approximations for maximizing submodular set functions*. Mathematical Programming.
[27] Matthew Richardson and Pedro Domingos. 2002. Mining knowledge-sharing sites for viral marketing. In *KDD*.
[28] Lior Seeman and Yaron Singer. 2013. Adaptive seeding in social networks. In *FOCS*. IEEE.
[29] Lichao Sun, Weiran Huang, Philip S Yu, and Wei Chen. 2018. Multi-Round Influence Maximization (Extended Version). *arXiv preprint arXiv:1802.04189* (2018).
[30] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: a martingale approach. In *SIGMOD*.
[31] Sharan Vaswani, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks V.S. Lakshmanan, and Mark Schmidt. 2017. Diffusion Independent Semi-Bandit Influence Maximization. In *ICML*.
[32] Chi Wang, Wei Chen, and Yajun Wang. 2012. Scalable influence maximization for independent cascade model in large-scale social networks. *DMKD* (2012).
[33] Zheng Wen, Branislav Kveton, Michal Valko, and Sharan Vaswani. 2017. Online influence maximization under independent cascade model with semi-bandit feedback. In *NIPS*.