Efficient approximation algorithms for adaptive influence maximization

Keke Huang · Jing Tang · Kai Han · Xiaokui Xiao · Wei Chen · Aixin Sun · Xueyan Tang · Andrew Lim

1 Introduction

The proliferation of online social networks such as Facebook and Twitter has motivated considerable research on viral marketing as an optimization problem. For example, an advertiser could provide a few individuals (referred to as “seed nodes”) in a social network with free product samples,
in exchange for them to spread the good words about the product, so as to create a large cascade of influence on other social network users via word-of-mouth recommendations. This phenomenon has been firstly formulated as Influence Maximization (IM) problem in [20], which aims to select a number of seed nodes to maximize the influence propagation created.

Formally, the input to IM consists of a social network \( G = (V, E) \), a budget \( k \), and an influence model \( M \). The influence model \( M \) captures the uncertainty of influence propagation in \( G \), and it defines a set of realizations, each of which represents a possible scenario of the influence propagation among the nodes in \( G \). The problem seeks to activate (i.e., influence) a seed set \( S \) of \( k \) nodes that can maximize the expected number of influenced individuals over all realizations.

A plethora of techniques have been proposed for IM [1,3,13,15,19,20,22,26–28,32–34,36,37]. Almost all techniques, however, require that the seed set \( S \) be decided before the influence propagation process, which means that they work in a “non-adaptive” manner. In other words, if an advertiser has \( k \) product samples, she would have to commit all samples to \( k \) chosen social network users before observing how they may influence other users. In practice, however, an advertiser could employ a more adaptive strategy to disseminate the product samples. For example, she may choose to give out half of the samples, and then wait for a while to find out which users are influenced; after that, she could examine the set of users that have not been influenced, and then disseminate the remaining samples to \( k/2 \) users that have a large influence on \( U \). This strategy is likely to be more effective than giving out all \( k \) samples all at once, since the dissemination of the second batch of products is optimized using the knowledge obtained from the first batch’s results.

In fact, the above adaptive approach has been applied in HEALER [39], a software agent deployed in practice since 2016, which recommends sequential intervention plans for homeless shelters. HEALER aims to raise awareness about HIV among homeless youth by maximizing the spread of awareness in the social network of the target population. It chooses people as the seed nodes, who are “activated” by participating the intervention plans for HIV. The choices of seed nodes are adaptive, i.e., they are selected in batches and the choice of a batch depends on the observed results of all previous batches.

Golovin and Krause [14] are the first to study IM under the adaptive setting, assuming that the \( k \) seed nodes are chosen in a sequential manner, such that the selection of the \((i+1)\)-th node is performed after the influence of the first \( i \) nodes has been observed. Specifically, they consider that (i) the social network conforms to a realization \( \phi \) that is generated by independently same every edge in graph \( G \) (according to the independent cascade model), but (ii) \( \phi \) is not known to the advertiser before the selection of the first seed node. Then, after the \( i \)-th seed node \( v_i \) is chosen, the part of \( \phi \) relevant to \( \{v_1, v_2, \ldots, v_i\} \) (i.e., the nodes that they can influence in \( \phi \)) is revealed to the advertiser, based on which she can (i) eliminate the realizations that contradict what she observes, and (ii) select the next seed node as one that has a large expected influence over the remaining realizations.

Golovin and Krause [14] propose a simple greedy algorithm for adaptive IM that returns a seed set \( S \) whose influence is at least \( 1 - 1/e \) of the optimum under the case that only one seed is selected in each batch (i.e., \( b = 1 \)). Nevertheless, the algorithm requires knowing the exact expected influence of every node, which is impractical since the computation of expected spread is \#P-hard in general [6,7]. Vaswani and Lakshmanan [38] extend Golovin et al.’s model by allowing selecting \( b \geq 1 \) seed nodes in each batch, and by accommodating errors in the estimation of expected spreads. Their method returns an \((1 - e^{-(1-1/e)^2/n})\)-approximation under this setting, where \( n \) is certain number bigger than 1. However, this relaxed approach is still impractical that in its requirement on the accuracy of expected spread estimation cannot be met by any existing algorithms (see Sect. 2.3 for a discussion).

To mitigate the above defects, there are two recent papers for adaptive IM, i.e., our preliminary work [16] and Sun et al.’s paper [31]. Han et al. [16] propose the first practical algorithm ADAPTIM. Meanwhile, Sun et al. [31] propose another approximation algorithm ADAIMM for a variant of the adaptive IM problem, referred to as Multi-Round Influence Maximization (MRIM). These two algorithms are claimed to provide the same worst-case approximation guarantee of \( 1 - e^{(1-1/e)(1-1/e)} \) with high probability, where \( e \in (0, 1) \) is a user-specified parameter. Unfortunately, both of their theoretical analyses on the approximation guarantee contain some gaps that invalidate their claims. We shall elaborate these misclaims in Sect. 5.

Contribution. Motivated by the deficiency of existing techniques and misunderstandings, we conduct an intensive study on the adaptive IM problem, and propose the first practical solution. Meanwhile, we derive a rigorous theoretical analysis that clarifies existing confusing points and lays a solid foundation for further study. Specifically, our contributions include the following.

First, we propose a novel randomized policy that can provide strong theoretical guarantees for the general adaptive stochastic maximization problem, which may be of independent interest. This new solution can be adopted in many other settings apart from adaptive IM, e.g., active learning [11], active inspection [17], optimal information gathering [10], which are special cases of adaptive stochastic maximization. In particular, our policy imposes far fewer constraints than the existing solutions [14], which are more applicable. The derivation of approximation results requires...
a non-trivial extension of the existing theoretical results on adaptive algorithms [14], and some new techniques like Azuma-Hoeffding inequality [25]. In addition, we propose a framework AdaptGreedy for adaptive IM that enables us to construct strong approximation solutions using existing non-adaptive IM methods as building blocks. In particular, we prove that AdaptGreedy achieves a worst-case approximation guarantee of $1 - e^{\rho_b(\varepsilon - 1)}$ with high probability when the number of adaptive rounds is reasonably large, where $\varepsilon \in (0, 1)$ is a user-specified parameter and $\rho_b = 1 - (1 - 1/b)^b$ is set by the batch size $b$. Moreover, we show that AdaptGreedy can also provide an expected approximation guarantee of $1 - e^{\rho_b(\varepsilon - 1)}$. Meanwhile, our analyses uncover some potential gaps in two recent works [16,31] and shed light on the future work of the adaptive IM problem.

Second, we conduct an in-depth analysis on how AdaptGreedy could be instantiated with the state-of-the-art non-adaptive IM algorithms. The overall approximation guarantee of AdaptGreedy relies on the expected approximation guarantee of the non-adaptive IM algorithm used by AdaptGreedy. However, existing non-adaptive IM algorithms do not benefit AdaptGreedy in this regard, as there is no known result on their expected approximation guarantees. Motivated by this fact, we develop a new non-adaptive IM method, EPIC, that provides an attractive expected approximation ratio by utilizing martingale stopping theorem [25]. We establish AdaptGreedy’s performance guarantee instantiated with EPIC.

Third, we conduct extensive experiments to test the performance of AdaptGreedy and EPIC, and the experimental results strongly corroborate the effectiveness and efficiency of our approach.

2 Preliminaries

2.1 IM and realization

Let $G = (V, E)$ be a social network with a node set $V$ and an edge set $E$, such that $|V| = n$ and $|E| = m$. We assume that the propagation of influence on $G$ follows the independent cascade (IC) model [20], in which each edge $(u, v)$ in $G$ is associated with a probability $p(u, v)$, and the influence propagation process is defined as a discrete-time stochastic process as follows. At timestamp 0, we activate a set $S$ of seed nodes. Then, at each subsequent timestamp $t$, each node $u$ that is newly activated at timestamp $t - 1$ has a chance to activate each of its neighbors $v$, such that the probability of activation equals $p(u, v)$. After that, $u$ stays active, but cannot activate any other nodes. The propagation process terminates when no node is newly activated at a certain timestamp, and the total number of nodes activated then is defined as the influence spread of $S$, denoted as $I_G(S)$. The vanilla influence maximization (IM) problem asks for a seed set $S$ of $k$ nodes that maximizes the expected value of influence spread $\E[I_G(S)]$.

As demonstrated in [20], the IC model also has an interpretation based on realization. Specifically, a realization $\phi$ represents a live-edge graph [20] generated by removing each edge $(u, v)$ in $G$ independently with $1 - p(u, v)$ probability. For example, Fig. 1 shows a social network and three of its realizations. We use $\Phi$ to denote a random realization. For any seed set $S$, let $I_\Phi(S)$ be the number of nodes in $\phi$ (including those in $S$) that can be reached from $S$ via a directed path starting from $S$, and $\E_\Phi[I_\Phi(S)]$ be the expectation over all realizations. It is shown in [20] that

$$\E_\Phi[I_\Phi(S)] = \E[I_G(S)].$$

In other words, if we are to address the vanilla IM problem, it suffices to identify a seed set $S$ whose expected spread over all realizations is the largest.

2.2 Adaptive IM

Suppose that the influence propagation on $G$ conforms to a realization $\phi$, i.e., for any seed set $S$, the nodes that it can influence are exactly the nodes that it can reach in $\phi$. The adaptive influence maximization (IM) problem [14] considers that $\phi$ is unknown in advance, but can be partially revealed after we choose some nodes as seeds. For example, consider the social network in Fig. 1a, and suppose that the realization is $\phi_1$, as shown in Fig. 1b. Assume that we choose $v_1$ as the first seed node. In that case, we can observe $v_1$’s influence on $v_2$ and $v_4$, since $v_1$ has two outgoing edges $(v_1, v_2)$ and $(v_1, v_4)$ in $\phi_1$. Similarly, we can observe $v_4$’s influence on $v_5$. In addition, we can also observe that $v_1$ (resp. $v_4$) cannot influence $v_3$ (resp. $v_6$), as $\phi_1$ does not contain an edge from $v_1$ to $v_3$ (resp. $v_4$ to $v_6$). Fig. 2a shows the results of the influence propagation from $v_1$, with each double-line (dashed-line) arrow denoting a successful (resp. failed) step of influence.

In general, after choosing a partial set $S'$ of seed nodes, we can learn all nodes that $S'$ can reach in $\phi$, as well as the out-edges of those nodes in $\phi$. This is referred to as the full-observation feedback model in [14]. This enables us to optimize the choices of the remaining seed nodes since we can focus on the nodes that have not been influenced by $S'$. For instance, consider that selecting another seed node based on the result in Fig. 2a. In that case, we can omit the nodes that have been influenced (i.e., $v_1, v_2, v_4$, and $v_5$), and focus on the subgraph induced by the remaining nodes, as shown in Fig. 2b. Based on this, we can choose $v_3$ as the second seed node, which yields the result in Fig. 2c, where we have 6 nodes influenced.
in total. In contrast, if we are to non-adaptively choose two seed nodes from the social network in Fig. 1a, we may end up choosing $v_1$ and $v_4$, in which case we would obtain the result in Fig. 2d when the realization is $\phi_1$ in Fig. 1b. In other words, we can only influence 4 nodes instead of 6 nodes.

Assume that we are to choose $k$ seed nodes in $r$ batches of equal size $b = k/r$, and that we are allowed to observe the influence propagation in $\phi$ for $r$ times in total, once after the selection of each batch. The adaptive IM problem asks for a seed selection policy that could generate the next seed set given the feedback of previous seed sets to maximize the expected influence spread over all realizations. Observe that when $b = k$ (i.e., $r = 1$), the problem degenerates to the vanilla IM problem.

We aim to develop algorithms for adaptive IM that provide non-trivial guarantees in terms of both accuracy (i.e., the expected influence of $\bigcup S_i$) and efficiency (i.e., the time required to identify $S_i$). We do not consider the “waiting time” required to observe the influence of a seed node batch $S_i$ before the selection of the next batch $S_{i+1}$, since it is independent of the algorithms used. That is, we target at helping the advertiser to identify $S_{i+1}$ as quickly as possible after the effects of $S_i$ have been observed.

Table 1 lists the notations that are frequently used in the remainder of the paper.

### 2.3 Existing solutions

The first solution to adaptive IM is by [14]. It assumes that $b = 1$ (i.e., each batch consists of only one seed node), and adopts a greedy approach as follows. Given $G$, it first identifies the node $v_1$ whose expected spread $\mathbb{E}[I_G(\{v_1\})]$ on $G$ is the largest, and selects it as the first seed. Then, it observes the nodes that are influenced by $v_1$ (which are in accordance to the realization $\phi_0$), and removes them from $G$.

Let $G_2$ denote the subgraph of $G$ induced by the remaining nodes. After that, for the $i$-th ($i > 1$) batch, it (i) selects the node $v_i$ with the maximum expected spread $\mathbb{E}[I_{G_i}(\{v_i\})]$ on $G_i$, (ii) observes the influence of $v_i$ on $G_i$, and then (iii) generates a new graph $G_{i+1}$ by removing from $G_i$ those nodes that are influenced by $v_i$. For convenience, we refer to $G_i$ as the $i$-th residual graph, and let $G_1 = G$.

Let $\text{OPT}_{k,b}$ denote the expected spread of the optimal solution to the adaptive IM problem parameterized with $k$ and $b$. Golovin et al. [14] show that the above greedy approach returns a solution whose expected spread is at least

---

**Table 1** Frequently used notations

| Notation | Description |
|----------|-------------|
| $G = (V, E)$ | A social network with node set $V$ and edge set $E$ |
| $n, m$ | The numbers of nodes and edges in $G$, respectively |
| $k$ | The total number of selected seed nodes |
| $b$ | The number of nodes selected in each batch |
| $G_i$ | The $i$-th residual graph |
| $n_i, m_i$ | The numbers of nodes and edges in $G_i$, respectively |
| $S_i$ | The seed set selected from $G_i$ |
| $S_i^o$ | The optimal seed set in $G_i$ |
| $\rho_b$ | Approximation guarantee for MAXCOVER with $\rho_b = 1 - (1 - 1/b)^b$. |
| $\text{OPT}_{k,b}$ | The optimal expected influence spread of $k$ seed nodes under the setting of selecting $b$ nodes in each batch |
| $\text{OPT}_{b}(G_i)$ | The optimal expected influence spread of $b$ seed nodes in $G_i$ |
| $I_G(S)$ | The number of nodes activated by $S$ in $G$ |
| $\text{Cov}_R(S)$ | The number of RR-sets in $R$ that overlap $S$ |
| $\text{F}_R(S)$ | The fraction of RR-sets in $R$ that overlap $S$ |
| $\mathbb{E}[I(S)]$ | The expected spread of seed set $S$ |
Finally, we propose a general framework adaptIM to address the adaptive IM problem.

3.1 Notations and definitions

Let $E$ be a finite set of items (e.g., a set of node sets), and $O$ be a set of possible states (e.g., the activation statuses of nodes). A realization is a function $\Phi : E \mapsto O$ mapping every item $e$ to a state $o$. We use $\Phi$ to denote a random realization. Let $p(\phi) := \Pr[\Phi = \phi]$ be the probability distribution over all realizations. We sequentially select an item $e$, and then observe its state $\Phi(e)$. Based on the observation, we would choose the next item and get to see its state, and so on. We use $\psi$, referred to as partial realization, to represent the relation such that $\psi := \{(e, o) : \Phi(e) = o\}$ for any $\psi \subseteq E \times O$. Let $\text{dom}(\psi)$ denote the domain of $\psi$ such that $\text{dom}(\psi) := \{e : \exists o, (e, o) \in \psi\}$. A partial realization $\psi$ is consistent with a realization $\phi$, referred to as $\phi \sim \psi$, if for every $e \in \text{dom}(\psi), \Phi(e) = \phi(e)$. Furthermore, we say $\psi \subseteq \psi'$, i.e., $\psi$ is a subrealization of $\psi'$, if there exists some $\phi$ such that $\phi \sim \psi$ and $\phi \sim \psi'$, and $\text{dom}(\psi) \subseteq \text{dom}(\psi')$.

A policy $\pi$ is an adaptive strategy for selecting items in $E$ based on current partial realization $\psi$. In this paper, we consider a randomized policy that selects items following certain distribution. To explicitly reveal the randomness of a randomized policy, we denote $\pi(\omega)$ as a random policy chosen from a set of all possible deterministic policies with respect to a random variable $\omega$. Intuitively, $\omega$ represents all random source of the randomized policy. In addition, let $\pi(\omega, \psi)$ be the item picked by policy $\pi(\omega)$ under partial realization $\psi$. We denote $E(\pi(\omega), \phi)$ as the set of items selected by $\pi(\omega)$ under realization $\phi$. We consider a utility function $f : 2^E \times O^E \mapsto \mathbb{R}_0$ depending on the picked items and their states. Then, the expected utility of a policy $\pi(\omega)$ is $f_{\text{avg}}(\pi(\omega)) := \mathbb{E}_{\omega}[f(E(\pi(\omega), \phi), \phi)]$. The goal of the adaptive stochastic maximization problem is to find a randomized policy $\pi^*$ such that

$$\pi^* \in \arg\max_{\pi} \mathbb{E}_{\omega}[f_{\text{avg}}(\pi(\omega))]$$

s.t. $|E(\pi(\omega), \phi)| \leq r$ for all $\omega$ and all $\phi$.

In addition, for any partial realization $\psi$, let $\Delta(e | \psi)$ and $\Delta(\pi(\omega) | \psi)$ denote the conditional marginal benefit of an item $e$ and a policy $\pi(\omega)$ conditioned on observing partial realization $\psi$, defined as

$$\Delta(e | \psi) := \mathbb{E}_{\phi}[f(\text{dom}(\psi) \cup \{e\}, \phi) \mid \Phi \sim \psi] - \mathbb{E}_{\phi}[f(\text{dom}(\psi), \phi) \mid \Phi \sim \psi],$$

$$\Delta(\pi(\omega) | \psi) := \mathbb{E}_{\phi}[f(\text{dom}(\psi) \cup E(\pi(\omega), \phi), \phi) \mid \Phi \sim \psi] - \mathbb{E}_{\phi}[f(\text{dom}(\psi), \phi) \mid \Phi \sim \psi].$$

(3)
We are now ready to introduce the notations of monotonicity and submodularity to the adaptive setting:

**Definition 1** (Adaptive Monotonicity) A function $f$ is adaptive monotone with respect to the realization distribution $p(\phi)$ if for all $\psi$ with $\Pr[\phi \sim \psi] > 0$ and all $e \in \mathcal{E}$, we have

$$\Delta(e \mid \psi) \geq 0.$$ 

**Definition 2** (Adaptive Submodularity) A function $f$ is adaptive submodular with respect to the realization distribution $p(\phi)$ if for all $\psi \subseteq \psi'$ and $e \in \mathcal{E} \setminus \text{dom}(\psi')$, we have

$$\Delta(e \mid \psi) \geq \Delta(e \mid \psi').$$ 

**Remark** Note that a deterministic policy is a special randomized policy. Meanwhile, the solution of any randomized policy is a convex combination of solutions of deterministic policies. Thus, the optimal solution of any randomized policy can always be achieved by some deterministic policy. As a consequence, for any randomized policy $\pi$ and any deterministic policy $\pi'$, it holds that $\max_{\pi} \mathbb{E}_\omega[f_{\text{avg}}(\pi(\omega))] = \max_{\pi'} f_{\text{avg}}(\pi').$

### 3.2 Adaptive greedy policy

A policy $\pi(\omega)$ is called an $\alpha$-approximate greedy policy if for all $\psi$, it always picks an item such that

$$\Delta(\pi(\omega), \psi \mid \psi) \geq \alpha \max_e \Delta(e \mid \psi).$$

Golovin and Krause [14] show that when the utility function $f$ is adaptive monotone and adaptive submodular, an $\alpha$-approximate greedy policy $\pi$ can achieve an approximation ratio of $(1 - e^{-\alpha})$ for the adaptive stochastic maximization problem, i.e., $\mathbb{E}_\omega[f_{\text{avg}}(\pi(\omega))] \geq (1 - e^{-\alpha})\mathbb{E}_\omega[f_{\text{avg}}(\pi^*(\omega))]$ for all policies $\pi^*$. However, in some applications, we would construct a randomized policy that may perform arbitrary worse (with low probability). For example, if the true value of $\Delta(e \mid \psi)$ is difficult to obtain, a policy maximizes an estimate of $\Delta(e \mid \psi)$ using a sampling method. For simplicity, we omit $\pi^*$ in $\mathbb{E}_\omega[f_{\text{avg}}(\pi(\omega))]$. However, even with some probability, even though very small, all the state-of-the-art IM algorithms may perform arbitrary worse. Such a randomized policy is not an $\alpha$-approximate greedy policy, for which Golovin and Krause’s theoretical results [14] are not applicable.

We consider a general version of randomized policy $\pi^g$ that can return an expected $\alpha_i$-approximate solution for the $i$-th item selection under every partial realization $\psi_{i-1}$, where $\psi_{i-1}$ represents a partial realization after we pick the first $(i - 1)$ items, i.e., for every $\psi_{i-1}$, $\mathbb{E}_\omega[\xi(\pi^g(\omega), \psi_{i-1})] \geq \alpha_i$. For a conventional version of $\pi^g$, one may set $\alpha_i = \alpha$ for every $i$, while for the general version of $\pi^g$, $\alpha_i$’s can be distinct.

Let $\Psi_i(\pi(\omega), \phi)$ represent a random partial realization after the policy $\pi(\omega)$ picks the first $i$ items under the realization $\phi$. For simplicity, we omit $\pi^g$ in $\Psi_i(\pi^g(\omega), \phi)$ when policy $\pi^g$ is used, i.e., $\Psi_i(\omega, \phi) := \Psi_i(\pi^g(\omega), \phi)$. Then, given any realization $\phi$ and any partial realization $\psi_{i-1}$ such that $\phi \sim \psi_{i-1}$, policy $\pi^g$ satisfies

$$\mathbb{E}_\omega[\xi(\pi^g(\omega), \psi_{i-1}) \mid \Psi_{i-1}(\omega, \phi) = \psi_{i-1}] \geq \alpha_i,$$

which describes that $\pi^g$ always returns an expected $\alpha_i$-approximate solution for the $i$-th item selection under every partial realization $\psi_{i-1}$ no matter what items are chosen by $\pi^g$ in the first $(i - 1)$ rounds.

To facilitate the analysis that follows, we define the notions of “policy truncation” and “policy concatenation”, which are conceptual operations performed by a policy.

**Definition 3** (Policy Truncation) For any adaptive policy $\pi$, the policy truncation $\pi_t$ denotes an adaptive policy that performs exactly the same as $\pi$, except that $\pi_t$ only selects the first $i$ items for any $i \leq r$.

**Definition 4** (Policy Concatenation) For any two adaptive policy $\pi$ and $\pi'$, the policy concatenation $\pi \oplus \pi'$ denotes an adaptive policy that first executes the policy $\pi$, and then
executes $\pi'$ from a fresh start as if any knowledge on the feedback obtained while running $\pi$ is ignored.

### 3.3.1 Expected approximation guarantee

The following theorem shows a concept of expected approximation guarantee for policy $\pi^{ag}$.

**Theorem 1** If $f$ is adaptive monotone and adaptive submodular, and $\pi^{ag}$ returns an expected $\alpha$-approximate solution for the $i$-th item selection under every partial realization $\Psi_{i-1}$, then the policy achieves an expected approximation guarantee of $1 - e^{-\alpha}$, where $\alpha = \frac{1}{r} \sum_{i} \alpha_i$ and $r$ is the total number of items selected, i.e., for all policies $\pi^*$, we have

$$E_{\omega}[f_{\text{avg}}(\pi^{ag}(\omega))] \geq (1 - e^{-\alpha})E_{\omega}[f_{\text{avg}}(\pi^*(\omega))].$$

(5)

Note that if a policy is an $\alpha$-approximate greedy policy, it must also be an expected $\alpha$-approximate greedy policy. Thus, our results generalize those given by Golovin and Krause [14]. The proof of Theorem 1 requires extensions of the theoretical results developed for adaptive stochastic maximization [14]. In the following, we first introduce some lemmas that are useful for proving Theorem 1.

**Lemma 1** For any deterministic adaptive policy $\pi$ and any $i \geq 0$, we have

$$f_{\text{avg}}(\pi_{i+1}) - f_{\text{avg}}(\pi_i) \leq E_{\phi}[\max_e \Delta(e | \pi_i(\pi, \Phi))].$$

**Proof** (Lemma 1) Let $p(\pi_i^*) := \text{Pr}[\Psi_i(\pi, \Phi) = \pi_i^*]$ be the probability of partial realization $\pi_i^*$ being observed after $\pi$ picks $i$ items over all realizations. We use $\pi_i^*$ to denote such a random partial realization with respect to the probability distribution $p(\pi_i^*)$. Then,

$$f_{\text{avg}}(\pi_{i+1}) - f_{\text{avg}}(\pi_i) = \mathbb{E}_{\omega}[f(\pi_{i+1}(\omega), \Phi) - f(\pi_i(\omega), \Phi)]$$

$$= \mathbb{E}_{\pi_i^*}[\mathbb{E}_{\omega}[f(\pi_{i+1}(\omega), \Phi) - f(\pi_i(\omega), \Phi) | \Phi \sim \pi_i^*]]$$

$$= \mathbb{E}_{\pi_i^*}[\Delta(\pi(\pi_i^*) | \psi)]$$

$$\leq \mathbb{E}_{\pi_i^*}[\max_e \Delta(e | \pi_i^*)]$$

$$= \mathbb{E}_{\pi_i^*}[\max_e \Delta(e | \pi_i(\pi, \Phi))].$$

where the inequality is because $\Delta(\pi(\pi_i^*) | \pi_i^*) \leq \max_e \Delta(e | \pi_i^*)$ for every $\pi_i^*$. □

**Lemma 2** Given any deterministic adaptive policy $\pi$ and any $i \leq j$, we have

$$\mathbb{E}_{\phi}[\max_e \Delta(e | \pi_i(\pi, \Phi))] \geq \mathbb{E}_{\phi}[\max_e \Delta(e | \pi_j(\pi, \Phi))].$$

**Proof** (Lemma 2) Again, let $\pi_i^*$ and $\pi_j^*$ denote random partial realizations with respect to the probability distribution $p(\pi_i^*)$ and $p(\pi_j^*)$, respectively. In addition, for every realization $\phi$ and any $i \leq j$, according to the nature of policy $\pi$, we have $\Psi_i(\pi, \phi) \subseteq \Psi_j(\pi, \phi)$. Thus, we can partition $\pi_i^*$ based on $\pi_j^*$. Then,

$$\mathbb{E}_{\phi}[\max_e \Delta(e | \pi_i(\pi, \Phi))]$$

$$= \mathbb{E}_{\pi_i^*}[\max_e \Delta(e | \pi_j^*)]$$

$$= \mathbb{E}_{\pi_i^*}[\mathbb{E}_{\phi}[\max_e \Delta(e | \pi_j^*) | \pi_i^* \subseteq \pi_j^*]]$$

$$= \mathbb{E}_{\pi_i^*}[\mathbb{E}_{\phi}[\Delta(e^*(\pi_j^*) | \pi_j^*) | \pi_i^* \subseteq \pi_j^*]]$$

$$\leq \mathbb{E}_{\pi_i^*}[\mathbb{E}_{\phi}[\Delta(e^*(\pi_j^*) | \pi_j^*) | \pi_i^* \subseteq \pi_j^*]]$$

$$= \mathbb{E}_{\pi_i^*}[\mathbb{E}_{\phi}[\max_e \Delta(e | \pi_i^*) | \pi_i^* \subseteq \pi_j^*]]$$

$$= \mathbb{E}_{\pi_i^*}[\max_e \Delta(e | \pi_i^*)]$$

$$= \mathbb{E}_{\pi_i^*}[\max_e \Delta(e | \pi_j(\pi, \Phi))].$$

where $e^*(\pi_j^*) := \arg \max_e \Delta(e | \pi_j^*)$ for each $\pi_j^*$. The first inequality is due to the adaptive submodularity of $f$, and the second inequality is because $\Delta(e^*(\pi_j^*) | \pi_j^*) \leq \max_e \Delta(e | \pi_j^*)$ for each $\pi_j^*$.

Using Lemmas 1 and 2, we can build a quantitative relationship between any policy $\pi$ and the optimal adaptive policy, as shown by Lemma 3.

**Lemma 3** For any deterministic adaptive policy $\pi$, any deterministic policy $\pi^*$ selecting $r$ items, and any $0 \leq i \leq r$, we have

$$f_{\text{avg}}(\pi^*) - f_{\text{avg}}(\pi_i) \leq r \cdot \mathbb{E}_{\phi}[\max_e \Delta(e | \pi_i(\pi, \Phi))].$$

**Proof** (Lemma 3) Each deterministic policy $\pi$ can be associated with a decision tree $T^{\pi}$ in a natural way. Each node in the decision tree is a partial realization $\psi$ such that the policy picks item $\pi(\psi)$ and the children of $\psi$ will be observed under respective realizations. Furthermore, each node $\psi$ is associated with a reward $r(\psi) := \text{Pr}[\Phi \sim \pi_i(\pi, \Phi) : \Delta(\pi(\psi) | \psi)]$ which is nonnegative due to the adaptive monotonicity of $f$, i.e., $\Delta(e | \psi) \geq 0$ for every $e$ and every $\psi$. Then, we can get that $f_{\text{avg}}(\pi) = \sum_{\psi \in T^{\pi}} r(\psi)$. In addition, it is easy to see that $T^{\pi} \subseteq T^{\pi \oplus \pi'}$. Thus, $f_{\text{avg}}(\pi \oplus \pi') - f_{\text{avg}}(\pi) = \sum_{\psi \in (T^{\pi \oplus \pi'} \setminus T^{\pi})} r(\psi) \geq 0$. Meanwhile, it is easy to verify that $f_{\text{avg}}(\pi \oplus \pi') = f_{\text{avg}}(\pi^* \ominus \pi')$, since $\pi \oplus \pi'$ and $\pi' \ominus \pi$ pick the same items under every realization.

For rotational convenience, let $\hat{\pi} := \pi_i \oplus \pi^*$ and $\hat{\pi}_{i+j} := \pi_i \oplus \pi^*_j$ for any $j \geq 0$. Thus, we have

$$f_{\text{avg}}(\pi^*) - f_{\text{avg}}(\pi_i) \leq f_{\text{avg}}(\hat{\pi}) - f_{\text{avg}}(\pi_i)$$

$$= \sum_{j=1}^{r} \left( f_{\text{avg}}(\hat{\pi}_{i+j}) - f_{\text{avg}}(\hat{\pi}_{i+j-1}) \right).$$
any randomized policy \( \pi \) be a randomized policy that returns an expected \( \alpha_i \)-approximate solution for the \( i \)-th item selection under every partial realization \( \psi_{i-1} \). For any \( 0 \leq i < r \) and any randomized policy \( \pi^* \), we have

\[
\mathbb{E}_\omega[f_{\text{avg}}(\pi^*(\omega)) - f_{\text{avg}}(\pi^*_r(\omega))] \\ \leq (1 - \frac{\alpha_i}{r}) \cdot \mathbb{E}_\omega[f_{\text{avg}}(\pi^*(\omega)) - f_{\text{avg}}(\pi^*_1(\omega))]
\]

(6)

where the expectation is over the randomness of \( \omega \).

**Proof** (Lemma 4) We first fix the randomness of \( \omega \). Then, \( \pi^*(\omega) \) and \( \pi^*_r(\omega) \) are deterministic policies. By Lemma 3, we have

\[
f_{\text{avg}}(\pi^*(\omega)) - f_{\text{avg}}(\pi^*_r(\omega)) \\ \leq r \cdot \mathbb{E}_\phi[\max_e \Delta(e \mid \psi_{i-1}(\omega, \Phi))].
\]

Taking the expectation over the randomness of \( \omega \) gives

\[
\mathbb{E}_\omega[f_{\text{avg}}(\pi^*(\omega)) - f_{\text{avg}}(\pi^*_r(\omega))] \\ \leq r \cdot \mathbb{E}_\phi[\mathbb{E}_\omega[\max_e \Delta(e \mid \psi_{i-1}(\omega, \Phi))]].
\]

(7)

On the other hand, by definition, we have

\[
\mathbb{E}_\omega[f_{\text{avg}}(\pi^*_1(\omega)) - f_{\text{avg}}(\pi^*_r(\omega))] \\ = \mathbb{E}_\omega[\mathbb{E}_\phi[\Delta(\pi^*_1(\omega) \mid \psi_{i-1}(\omega, \Phi))]] \\ = \mathbb{E}_\phi[\mathbb{E}_\omega[\Delta(\pi^*_1(\omega) \mid \psi_{i-1}(\omega, \Phi))]].
\]

(8)

In addition, for any given realization \( \phi \), let \( \psi^\phi_{i-1} \) be the random partial realization following the probability distribution \( p(\psi^\phi_{i-1}) := \text{Pr}[\psi_{i-1}(\omega, \phi) = \psi^\phi_{i-1}] \) over the randomness of \( \omega \). Then,

\[
\mathbb{E}_\omega[\Delta(\pi^*_1(\omega) \mid \psi_{i-1}(\omega, \phi))] \\ = \mathbb{E}_{\psi^\phi_{i-1}}[\mathbb{E}_\omega[\Delta(\pi^*_1(\omega) \mid \psi^{\psi^\phi_{i-1}}_{i-1}(\omega, \phi) = \psi^\phi_{i-1})]].
\]

The first inequality is due to the adaptive monotonicity of \( f \) as discussed above. The second inequality is by Lemma 1 while the third inequality is by Lemma 2. The final equality is because \( \tilde{\pi}_i = \pi_i \).

Then, we are able to establish a relationship between our proposed randomized policy \( \pi^{\text{ag}} \) and any randomized policy in the following lemma.

**Lemma 4** Let \( \pi^{\text{ag}} \) be a randomized policy that returns an expected \( \alpha_i \)-approximate solution for the \( i \)-th item selection under every partial realization \( \psi_{i-1} \). For any \( 0 \leq i < r \) and any randomized policy \( \pi^* \), we have

\[
\mathbb{E}_\omega[f_{\text{avg}}(\pi^*(\omega)) - f_{\text{avg}}(\pi^*_1(\omega))] \\ \leq r \cdot \mathbb{E}_\phi[\mathbb{E}_\omega[\max_e \Delta(e \mid \psi_{i-1}(\omega, \Phi))]].
\]

(9)

Rearranging it completes the proof.

Finally, we are ready to prove Theorem 1.

**Proof** (Theorem 1) Note that for any \( x \) such that \( 0 \leq x \leq 1 \), we have \( 1 - x \leq e^{-x} \). Therefore, recursively applying Lemma 4 gives

\[
\mathbb{E}_\omega[f_{\text{avg}}(\pi^*(\omega)) - f_{\text{avg}}(\pi^*_1(\omega))] \\ \leq e^{-\alpha_i/r} \cdot \mathbb{E}_\omega[f_{\text{avg}}(\pi^*(\omega)) - f_{\text{avg}}(\pi^*_r(\omega))] \\ \leq \cdots \\ \leq e^{-\sum_{i=1}^r \alpha_i/r} \cdot \mathbb{E}_\omega[f_{\text{avg}}(\pi^*(\omega)) - f_{\text{avg}}(\pi^*_0(\omega))] \\ = e^{-\alpha} \cdot \mathbb{E}_\omega[f_{\text{avg}}(\pi^*(\omega))].
\]

Rearranging it completes the proof.

\( \square \)

### 3.3.2 Worst-case approximation guarantee

In what follows, we derive another concept of worst-case approximation guarantee for policy \( \pi^{\text{ag}} \). To begin with, we first provide a random approximation guarantee for \( \pi^{\text{ag}} \) as follows.

**Lemma 5** Let \( X_i(\omega) \) be the overall random approximation for the \( i \)-th item selection achieved by \( \pi^{\text{ag}}(\omega) \) with respect to \( \omega \), i.e.,

\[
X_i(\omega) := \frac{\mathbb{E}_\phi[\Delta(\pi^*_1(\omega) \mid \psi_{i-1}(\omega, \Phi))]}{\mathbb{E}_\phi[\max_e \Delta(e \mid \psi_{i-1}(\omega, \Phi))]}.
\]

Then, \( \pi^{\text{ag}}(\omega) \) achieves a random approximation guarantee of \( 1 - e^{-X(\omega)} \), where \( X(\omega) = \frac{1}{r} \sum_{i=1}^r X_i(\omega) \).

\( \square \) Springer
Proof (Lemma 5) By definition, we have
\[
\begin{align*}
    f_{\text{avg}}(\pi^*_{i}(\omega)) - f_{\text{avg}}(\pi^*_{i-1}(\omega)) & = X_i(\omega) \cdot \mathbb{E}_{\Phi}\left[\max_{\varepsilon} \Delta(\varepsilon | \Psi_{i-1}(\omega, \Phi))\right] \\
    & = X_i(\omega) \cdot \mathbb{E}_{\Phi}\left[\Delta(\pi^*_{i}(\omega) | \Psi_{i-1}(\omega, \Phi))\right]
\end{align*}
\]
Together with Lemma 3, we have
\[
\begin{align*}
    f_{\text{avg}}(\pi^*_{i}(\omega)) - f_{\text{avg}}(\pi^*_{i-1}(\omega)) & \leq \left(1 - \frac{X_i(\omega)}{\Delta}\right) \cdot \left(f_{\text{avg}}(\pi^*_{i}(\omega)) - f_{\text{avg}}(\pi^*_{i-1}(\omega))\right).
\end{align*}
\]
Using a similar argument for the proof of Theorem 1 immediately concludes Lemma 5.

Note that in Lemma 5, for any given realization \(\phi\), the conditional expected marginal benefit of policy \(\pi^*_{i}(\omega)\) based on \(\Psi_{i-1}(\omega, \phi)\) equals to that of item \(\pi^*_{i}(\omega, \Psi_{i-1}(\omega, \phi))\).

The random approximation guarantee in Lemma 5 is crucial in providing worst-case theoretical guarantees.

To this end, a simple and intuitive idea is to show that \(X_i(\omega) \geq \alpha_i\) with high probability for every \(i\). Then, by a union bound for \(r\) rounds of item selection, we can obtain a worst-case approximation guarantee. However, it is hard to derive such a non-trivial worst-case approximation guarantee, as the probability that \(X_i(\omega) < \alpha_i\) could be large even though the probability of \(\xi(\pi^*_{i}(\omega, \Psi_{i-1})) < \alpha_i\) on any given \(\psi_{i-1}\) is small, where \(\xi(\pi^*_{i}(\omega, \Psi_{i-1})) = \Delta(\pi^*_{i}(\omega) | \Psi_{i-1})\).

To explain the number of possible \(\psi_{i-1}\)‘s can be as large as an exponential scale size, e.g., \(O(2^m)\) realizations of influence propagation where \(m\) is the number of edges in \(G\). Once there exists one instance of \(\psi_{i-1}\) such that \(\xi(\pi^*_{i}(\omega, \Psi_{i-1})) < \alpha_i\), it is possible that \(X_i(\omega) < \alpha_i\).

In other words, to ensure that \(X_i(\omega) \geq \alpha_i\), one sufficient way is to guarantee that \(\xi(\pi^*_{i}(\omega, \Psi_{i-1})) \geq \alpha_i\) for every \(\psi_{i-1}\). However, such a requirement is too stringent to satisfy. Unfortunately, two recent papers [16] and [31] claim that if \(\Pr[\xi(\pi^*_{i}(\omega, \Psi_{i-1})) < \alpha_i] \leq \delta_i\) for every \(\psi_i\), then \(\Pr[X_i(\omega) < \alpha_i] \leq \delta_i\). This misclaim makes their approximation guarantees invalid. More details are presented in Sect. 5.

On the other hand, the strategy that demands every \(X_i(\omega) \geq \alpha_i\) with high probability is also overly conservative. For example, suppose that there exists one \(X_j(\omega)\) satisfying \(X_j(\omega) < \alpha_j\), i.e., it fails to achieve the overall \(\alpha_j\)-approximation in the \(j\)-th item selection. Even in that case, the overall approximation ratio of \(\pi^*_{i}(\omega)\) could still be better than \(1 - e^{-\Delta}\), as long as there exists another \(X_i(\omega)\) satisfying \(X_i(\omega) \geq \alpha_i + \alpha_j - X_j(\omega)\). In other words, the deficiency of one round can be compensated as long as there exists other rounds whose quality is above the bar by a sufficient margin.

Formally, as the approximation ratio \(X_i(\omega)\) in each round of \(\pi^*_{i}(\omega)\) is a random variable, the overall approximation guarantee of \(\pi^*_{i}(\omega)\), namely, \(1 - e^{-X_i(\omega)}\), depends on the mean of all \(r\) variables. Intuitively, when \(r\) is sizable, \(X(\omega) = \frac{1}{r} \sum_{i=1}^{r} X_i(\omega)\) should be concentrated to its expectation, i.e., \(\mathbb{E}[X(\omega)]\). Note that \(\Pr[X(\omega) \geq \alpha_i] \geq \alpha_i\) holds for every \(\psi_{i-1}\), which is exactly the requirement of \(\pi^*_{i}(\omega)\) for each round of item selection. That is, instead of formulating the approximation ratio of \(\pi^*_{i}(\omega)\) based on the worst-case guarantee of each selected item, we might derive it based on each selected item’s expected approximation ratio.

To make the above idea work, the distance between \(X(\omega)\) and its expectation \(\mathbb{E}[X(\omega)]\) is to be bounded with high probability. However, there is a challenge that we need to address. As the selection of the \(i\)-th item is dependent on the results of the first \((i-1)\) items, the random variables \(X_1(\omega), X_2(\omega), \ldots, X_r(\omega)\) are correlated, making it rather non-trivial to derive concentration results for \(\frac{1}{r} \sum_{i=1}^{r} X_i(\omega)\).

We circumvent this issue with a theoretical analysis by leveraging Azuma-Hoeffding inequality for martingales [25].

Definition 5 (Martingale [25]) A sequence of random variables \(Z_0, Z_1, \ldots\) is a martingale with respect to the sequence \(Y_0, Y_1, \ldots\) if, for all \(i \geq 0\), the following conditions hold:

\[
\begin{align*}
    & - Z_i \text{ is a function of } Y_0, Y_1, \ldots, Y_i; \\
    & - \mathbb{E}[|Z_i|] < \infty; \\
    & - \mathbb{E}[Z_{i+1} | Y_0, Y_1, \ldots, Y_i] = Z_i.
\end{align*}
\]

Lemma 6 (Azuma-Hoeffding Inequality [25]) Let \(Z_0, Z_1, \ldots\) be a martingale with respect to the sequence of random variables \(Y_0, Y_1, \ldots\) such that

\[
B_i \leq Z_i - Z_{i-1} \leq B_i + c_i
\]

for some constants \(c_i\) and for some random variables \(B_i\) that may be functions of \(Y_0, \ldots, Y_{i-1}\). Then, for any \(r \geq 0\) and any \(\lambda > 0\),

\[
\Pr[Z_r \geq Z_0 + \lambda] \leq e^{-2\lambda^2/\sum_{i=1}^{r} c_i^2}.
\]

Based on the above Azuma-Hoeffding inequality, we provide a concentration bound for possibly correlated random variables as follows.

Corollary 1 Let \(Y_1, \ldots, Y_r\) be any sequence of random variables and \(Y'_r\) be a function of \(Y_1, \ldots, Y_r\) satisfying \(Y'_r \leq \beta\) and \(\mathbb{E}[Y'_r | Y_1, \ldots, Y_{r-1}] \leq \gamma_i\) for every \(i = 1, \ldots, r\). Then, we have

\[
\Pr\left[\sum_{i=1}^{r} Y'_i \geq \sum_{i=1}^{r} \gamma_i + \sqrt{r} \beta \lambda\right] \leq e^{-\lambda^2/2}. \tag{9}
\]

Proof (Corollary 1) Let \(Z_0 = Y'_0 = 0\) and for every \(i = 1, \ldots, r\),

\[
Z_i := \sum_{j=1}^{i} (Y'_j - \mathbb{E}[Y'_j | Y_1, \ldots, Y_{j-1}]).
\]
Then, it is easy to verify that $\mathbb{E}[|Z_i|] < \infty$ and $\mathbb{E}[Z_i \mid Y_1, \ldots, Y_{i-1}] = Z_{i-1}$, which indicates that $Z_i$ is a martingale. In addition, let $B_i := -\beta - \mathbb{E}[Y_i \mid Y_1, \ldots, Y_{i-1}]$. As $|Y_i| \leq \beta$, we can get that $Z_i - Z_{i-1} = Y_i - \mathbb{E}[Y_i \mid Y_1, \ldots, Y_{i-1}]$ is in the range of $[B_i, B_i + 2\beta]$. Thus, according to Lemma 6,

$$\Pr[Z_r \geq \sqrt{r}\beta\lambda] \leq e^{-2\beta^2\lambda^2/\sum_{i=1}^{r}(2\beta)^2} = e^{-\lambda^2/2}. (10)$$

On the other hand, as $\mathbb{E}[Y_i \mid Y_1, \ldots, Y_{i-1}] \leq \gamma_i$ for every $i = 1, \ldots, r$, we have $Z_r \geq \sum_{i=1}^{r}(Y_i - \gamma_i)$. As a consequence

$$\Pr[\sum_{i=1}^{r}(Y_i - \gamma_i) \geq \sqrt{r}\beta\lambda] \leq \Pr[Z_r \geq \sqrt{r}\beta\lambda]. (11)$$

Combining (10) and (11) completes the proof. \qed

Recall that we have shown in Lemma 5 that the approximation guarantee of $\pi^{\text{ag}}$ is determined by the summation of the approximation factors, i.e., $X(\omega) = 1/2 \sum_{i=1}^{r}X_i(\omega)$, and these approximation factors could be correlated. According to Corollary 1, we can get a bound on their summation, based on which we can derive the overall approximation guarantee as follows.

**Theorem 2** Without loss of generality, suppose that policy $\pi^{\text{ag}}$ returns $X(\pi^{\text{ag}}(\omega), \psi)$-approximate solution satisfying $c_1 \leq \xi(\pi^{\text{ag}}(\omega), \psi) \leq c_2$ for every $\omega$ and $\psi$. For any given $\delta \in (0, 1)$, let $\alpha' = \frac{1}{r} \sum_{i=1}^{r} \alpha_i - (c_2 - c_1) \cdot \sqrt{1/(2\gamma)} \cdot \ln(1/\delta)$. If $f$ is adaptive monotone and adaptive submodular, and $\pi^{\text{ag}}$ returns an expected $\alpha_i$-approximate solution for the $i$-th item selection under every partial realization $\pi_j(\omega)$, then $\pi^{\text{ag}}$ achieves the worst-case approximation ratio $1 - e^{-\alpha'}$ with a probability of at least $1 - \delta$.

**Proof** (Theorem 2) For every algorithm randomness $\omega$ and every realization $\phi$, as defined before, $\Psi_j(\omega, \phi)$ represents the partial realization corresponding to the first $i$-steps of running the adaptive greedy policy $\pi^{\text{ag}}(\omega)$ on realization $\phi$, for $i = 1, 2, \ldots, r$. Then $\Psi_j(\omega, \cdot)$ is a mapping from all realizations to partial realizations that has $i$ items in the domain. Let $A_j$ be any fixed mapping from all realizations to partial realizations that has $i$ items in the domain and is consistent with the corresponding realization, i.e., $\phi \sim A_i(\phi)$ for all $\phi$ and $|\text{dom}(A_i(\phi))| = i$.

Let $\Omega_{i-1}$ be the distribution of $\omega$ conditional on $\Psi_j(\omega, \cdot) = A_j$ for every $j < i$. That is, $\Omega_{i-1}$ is the probability subspace in which the adaptive greedy policy $\pi^{\text{ag}}$ generates the first $i - 1$ steps exactly according to $A_1, A_2, \ldots, A_{i-1}$. For any $\omega$ sampled from $\Omega_{i-1}$, by the above definition, we have that

$$\Pr[X(\omega) \leq \alpha'] = \Pr\left[X(\omega) \leq \frac{1}{r} \sum_{i=1}^{r} \alpha_i - (c_2 - c_1) \cdot \sqrt{\frac{1}{2\gamma}} \cdot \ln(1/\delta)\right] \leq e^{-\ln 2 - \frac{1}{2} \ln 2} = \delta,
$$

which completes the proof. \qed

Note that the worst-case approximation ratio of $\pi^{\text{ag}}$ (Theorem 2) is worse than its expected approximation guarantee (Theorem 1), where the overall approximation factor of the latter is larger by an additive factor of $(c_2 - c_1) \cdot \sqrt{1/(2\gamma)} \cdot \ln(1/\delta)$ than the former. Furthermore, as $\pi^{\text{ag}}$ is a $c_1$-approximate greedy policy, according to Golovin and Krause [14], $\pi^{\text{ag}}$ achieves an approximation ratio of

$$E_{\omega}[X_i(\omega) \mid \omega \sim \Omega_{i-1}]$$

for every realization $\omega$, $\Psi_j(\omega, \phi) = A_j(\phi)$. Note that, to be precise, we would include $A_1, A_2, \ldots, A_{i-1}$ in the notation $\Omega_{i-1}$, such as $\Omega_{i-1}(A_1, A_2, \ldots, A_{i-1})$, but for simplicity we choose the shorter notation. Then we have

$$E_{\omega}[X_i(\omega) \mid \omega \sim \Omega_{i-1}] = E_{\omega}\left[\frac{\Delta(\pi^{\text{ag}}(\omega) \mid \Psi_j(\omega, \Phi))}{\max_e \Delta(e \mid \Psi_j(\omega, \Phi))} \mid \omega \sim \Omega_{i-1}\right]$$

where the inequality is by the requirement of $\pi^{\text{ag}}$ in (4).

Note that by definition, $E_{\omega}[X_i(\omega) \mid \omega \sim \Omega_{i-1}]$ represents $E_{\omega}[X_i(\omega) \mid \Psi_j(\omega, \cdot) = A_j, \forall j < i]$. Inequality (12) means that for any fixed mappings $A_1, \ldots, A_{i-1}$, $E_{\omega}[X_i(\omega) \mid \Psi_j(\omega, \cdot) = A_j, \forall j < i] \geq \alpha_i$. Omitting $A_1, \ldots, A_{i-1}$, we have $E_{\omega}[X_i(\omega) \mid \Psi_j(\omega, \cdot), \forall j < i] \geq \alpha_i$.

Next, by letting $Y_i(\omega) = 1/2 - (X_i(\omega) - c_1)/(c_2 - c_1)$, we have $E_{\omega}[Y_i(\omega) \mid \Psi_j(\omega, \cdot), \forall j < i] \leq 1/2 - (\alpha_i - c_1)/(c_2 - c_1)$. Meanwhile, it is easy to obtain that $c_1 \leq X_i(\omega) \leq c_2$ as $c_1 \leq \xi(\pi^{\text{ag}}(\omega), \psi) \leq c_2$ for every $\omega$ and $\psi$. Thus, we also have $|Y_i|^\phi \leq 1/2$.

Hence, by treating $\Psi_j(\omega, \cdot)$ as the random variable $Y_j$ in Corollary 1, we can apply Corollary 1 and obtain

$$Pr[X(\omega) \leq \alpha'] \leq Pr\left[X(\omega) \leq \frac{1}{r} \sum_{i=1}^{r} \alpha_i - (c_2 - c_1) \cdot \sqrt{\frac{1}{2\gamma}} \cdot \ln(1/\delta)\right] \leq e^{-\frac{1}{2} \ln 1/\delta} = \delta,$$

which completes the proof. \qed
(1 − e^{−ε_1}). Thus, the worst-case approximation ratio is meaningful only when \( \alpha' \geq c_1 \), i.e., \( \delta \geq e^{−(L_i \log(ω))} \).

### 3.4 Solution framework for adaptive IM

The adaptive IM under the IC model satisfies the adaptive monotonicity and adaptive submodularity [14]. Based on the expected \( \alpha \)-approximate greedy policy \( \pi^{ag} \), we propose a general framework ADAPTGREEDY (i.e., Algorithm 1) upon which we can build specific algorithms with seed selection algorithms to address the adaptive IM problem. At the first glance, ADAPTGREEDY may seem similar to Vaswani and Lakshmanan’s method [38], since both techniques (i) adaptively select seed nodes in \( r \) batches and (ii) do not require exact computation of expected spreads. However, there is a crucial difference between the two: Vaswani and Lakshmanan’s method requires that the expected spread of every node set should be estimated with a small fixed relative error with respect to its own expectation, whereas ADAPTGREEDY just requires an expected approximation ratio of \( \rho_b(1 − ε_i) \) with respect to \( \text{OPT}_b(G_i) \) (Line 5 in Algorithm 1), where \( \text{OPT}_b(G_i) \) denotes the maximum expected spread of any size-\( b \) seed set on \( G_i \) and \( \rho_b = 1 − (1 − 1/b)^b \). Note that \( \rho_b \) is the approximation ratio achieved by a greedy algorithm for MAXCOVER (which is a building block for IM), and this factor cannot be further improved by any polynomial time algorithm unless \( \text{P} = \text{NP} \) [12]. The error requirement of ADAPTGREEDY is much more lenient than that of Vaswani and Lakshmanan’s method, and it can be achieved by several state-of-the-art solutions [26,33,36,37] for vanilla influence maximization, i.e., it admits practical implementations.

In addition, ADAPTGREEDY is flexible in that it allows each batch of seed nodes \( S_i \) to be selected with different approximation guarantee \( \rho_b(1 − ε_i) \), whereas the existing solutions (e.g., [14]) for adaptive IM require that all seed sets \( S_1, \ldots, S_r \) should be processed with identical accuracy assurance. Therefore, ADAPTGREEDY is a general framework for the adaptive IM problem. According to Theorem 1 and Theorem 2, ADAPTGREEDY can provide the following theoretical guarantees.

**Theorem 3** If ADAPTGREEDY returns an expected \( \rho_b(1 − ε_i) \)-approximate solution in the \( i \)-th batch of seed selection, then it achieves an expected approximation guarantee of \( 1 − e^{\rho_b(ε'_i − 1)} \), where \( ε_i = \frac{1}{r} \sum_{i=1}^{r} ε_i, \rho_b = 1 − (1 − 1/b)^b, r \) is the number of batches and \( b \) is the batch size.

Meanwhile, ADAPTGREEDY also achieves a worst-case approximation guarantee of \( 1 − e^{\rho_b(ε'_i − 1)} \) with a probability of at least \( 1 − δ, \) where \( ε'_i = \frac{1}{r} \sum_{i=1}^{r} ε_i + \sqrt{1/(2r) \cdot \ln(1/δ)}. \)

### 4 Instantiations of ADAPTGREEDY

In this section, we first present a naive instantiation of ADAPTGREEDY using the state-of-the-art non-adaptive IM algorithms. To utilize the notion of expected approximation ratio for each batch of seed selection, we then design a new non-adaptive IM algorithm EPIC. Finally, we analyze the approximation guarantees and time complexity of EPIC and ADAPTGREEDY instantiated with EPIC respectively.

#### 4.1 Instantiation using existing algorithms

As shown in Algorithm 1, ADAPTGREEDY requires identifying a random size-\( b \) seed set \( S_i(ω) \) with respect to the randomness\(^2\) of \( π^{ag}(ω) \) from the \( i \)-th residual graph \( G_i \), such that

\[
\mathbb{E}_ω[\mathbb{E}[I_{G_i}(S_i(ω))]] ≥ \rho_b(1 − ε_i)\text{OPT}_b(G_i),
\]

where \( \mathbb{E}[I_{G_i}(S_i(ω))] \) is the expected spread of \( S_i(ω) \) on \( G_i \) and its expectation \( \mathbb{E}_ω[\cdot] \) is over the internal randomness of the algorithm, \( \text{OPT}_b(G_i) \) is the maximum expected spread of any size-\( b \) seed set on \( G_i \). For brevity, in the rest of the paper, we use \( S_i(ω) \) to represent a random set \( S_i(ω) \) obtained by a randomized policy \( π^{ag}(ω) \) from the \( i \)-th residual graph \( G_i \).

We observe that such a seed set \( S_i \) can be obtained by applying the state-of-the-art algorithms (e.g., [26,33,36,37]) for vanilla influence maximization (IM) on \( G_i \). In particular, these algorithms are randomized, and they provide a worst-case approximation guarantee as follows: given a seed set size \( b \), a relative error threshold \( ε'_i \) and a failure probability \( δ_i \), they output a size-\( b \) seed set \( S_i \) in \( G_i \) whose expected spread is \( \rho_b(1 − ζ(ω, G_i)) \) times the maximum expected spread of any size-\( b \) seed set on \( G_i \), such that \( ζ(ω, G_i) ≤ ε'_i \) with at least \( 1 − δ_i \) probability. Thus, we obtain that

\[
\mathbb{E}_ω[ζ(ω, G_i)] ≤ ε'_i (1 − δ_i) + δ_i.
\]

\(^2\) Usually, the random source \( ω \) indicates sampling for IM, e.g., reverse influence sampling [3].
To ensure (13), for each pair of \((\varepsilon_i', \delta_i)\), let \(\delta_i\) be a sufficient small value and \(\varepsilon_i' := (\varepsilon_i - \delta_i)/(1 - \delta_i) \approx \varepsilon_i\). According to Theorem 3, such an instantiation of ADAPTGREEDY yields an expected (resp. worst-case) approximation ratio of \(1 - \varepsilon_0(b^{\varepsilon - 1})\) (resp. \(1 - \varepsilon_0(b^{\varepsilon - 1})\)) where \(\varepsilon = \frac{1}{r} \sum_{i=1}^r \varepsilon_i\) (resp. \(\varepsilon' = \frac{1}{r} \sum_{i=1}^r \varepsilon_i + \sqrt{1/2r \cdot \ln(1/\delta)}\)).

But how efficient is the above instantiation? To answer this question, we need to investigate the time complexity of the vanilla IM algorithms in [37]. The theoretical analysis in [37] shows that if we are to achieve \((\rho_0 - \varepsilon'_i)\)-approximation on \(G_i\) with at least \(1 - \delta_i\) probability, then the expected computation cost is \(O((b \log n_i + \log \frac{1}{\varepsilon'_i})(m_i + n_i)/\varepsilon'_i^2)\), where \(n_i\) and \(m_i\) denote the number of nodes and edges in \(G_i\), respectively. Since \(n_i \leq n\) and \(m_i \leq m\), the expected time required to process \(G_i\) is \(O((b \log n + \log \frac{1}{\varepsilon'}) (m + n)/\varepsilon^2)\). As such, all \(r\) batches of seed nodes can be identified in \(O((b \log n + \log \frac{1}{\varepsilon'}) (m + n)/\varepsilon^2)\) expected time. By setting a pair of parameters \((\varepsilon_i', \delta_i)\) in the vanilla IM algorithms as \(\varepsilon_i' = (\varepsilon_i - \delta_i)/(1 - \delta_i)\), we can achieve an expected approximation ratio of \(\rho_0(1 - \varepsilon)\) in the \(i\)-th batch. This shows that the total expected time complexity for achieving the final expected (resp. worst-case) approximation ratio of \(1 - \varepsilon_0(b^{\varepsilon - 1})\) (resp. \(1 - \varepsilon_0(b^{\varepsilon - 1})\)) is \(O((\varepsilon_i') (m + n)/\varepsilon_i^2)\).

**Rationale for an Improved Approach.** The aforementioned instantiation of ADAPTGREEDY is straightforward and intuitive, but is far from optimized in terms of its approximation guarantee. To explain, recall that it requires each seed set \(S_i\) to achieve \(\rho_0(1 - \varepsilon'_i)\)-approximation on \(G_i\) with probability at least \(1 - \delta_i\), based on which it provides an overall expected approximation ratio of \(1 - \varepsilon_0(b^{\varepsilon - 1})\) with \(\varepsilon = \frac{1}{r} \sum_{i=1}^r \varepsilon_i\) where \(\varepsilon_i = \varepsilon'_i/(1 - \delta_i) + \delta_i \approx \varepsilon'_i\). In other words, it imposes a stringent worst-case approximation guarantee on each seed set \(S_i\). This, however, might be overly conservative. Intuitively, the expected approximation error factor \(E_{\omega}[\zeta(\omega, G_i)]\) should be much smaller than the naive upper bound \(\varepsilon_i\) deduced from the worst-case approximation. To the best of our knowledge, there is no known result for vanilla IM with tight expected approximation guarantees. This motivates us to develop a vanilla IM algorithm tailored for ADAPTGREEDY, as we show in the following section.

### 4.2 IM algorithm with expected approximation

As discussed in Sect. 4.1, the existing IM algorithms provide only a worst-case approximation guarantee, i.e., the relative error factor \(\zeta(\omega, G_i)\) is no more than the input threshold \(\varepsilon'_i\) with high probability. To optimize the performance of ADAPTGREEDY, we are in need of one non-adaptive IM algorithm with expected approximation guarantee \(\rho_0(1 - E_{\omega}[\zeta(\omega, G_i)])\) such that \(E_{\omega}[\zeta(\omega, G_i)]\) has a tighter bound. In what follows, we present a new non-adaptive IM algorithm, referred to as EPIC, that returns a solution with expected approximation guarantee. To this end, we first introduce the concept of reverse reachable sets (RR-sets) [3], which is the basis of our algorithm.

**RR-Sets.** In a nutshell, RR-sets are subgraph samples of \(G\) that can be used to efficiently estimate the expected spreads of any given seed sets. Specifically, a random RR-set of \(G\) is generated by first selecting a node \(v \in V\) uniformly at random, and then taking the nodes that can reach \(v\) in a random graph generated by independently removing each edge \((u, v) \in E\) with probability \(1 - p(u, v)\). If a seed node set \(S\) has large expected influence spread, then the probability that \(S\) intersects with a random RR-set is high, as shown in the following equation [3]:

\[
\mathbb{E}[I_G(S)] = n \cdot \Pr[R \cap S \neq \emptyset],
\]

where \(R\) is a random RR-set. This result suggests a simple method for estimating the expected influence spread of any node set \(S\): we can use a set \(R\) of random RR-sets to estimate the value of \(\Pr[R \cap S \neq \emptyset]\) and hence \(\mathbb{E}[I_G(S)]\). In particular, let \(Cov_R(S)\) denote the number of RR-sets in \(R\) that overlap \(S\). Then the value of \(\mathbb{E}[I_G(S)]\) can be unbiasedly estimated by \(n \cdot F_R(S)\), where

\[
F_R(S) = \frac{Cov_R(S)}{|R|}.
\]

By the law of large numbers, \(n \cdot F_R(S)\) should converge to \(\mathbb{E}[I_G(S)]\) when \(|R|\) is sufficiently large, which provides a way to estimate \(\mathbb{E}[I_G(S)]\) to any desired accuracy level. However, due to the cost of generating RR-sets, there is a tradeoff between accuracy and efficiency in any algorithms using RR-set sampling.

**The. EPIC Algorithm** Algorithm 2 shows the pseudo-code of our EPIC algorithm, which borrows the idea from the OPIM-C algorithm [33] via (i) starting from a small number of RR-sets and (ii) iteratively increasing the RR-set number until a satisfactory solution is identified. The key difference between the two algorithms lies in the way that they compute the upper bound \(F'(S')\), i.e., the fraction of RR-sets that overlap \(S'\) to any desired accuracy level.

### References

1. Expected approximation for influence maximization.

@article{Huang2014, 
author = {K. Huang et al.}, 
title = {expected approximation for influence maximization}, 
journal = {arXiv preprint arXiv:1403.4314}, 
year = {2014}, 
url = {http://arxiv.org/abs/1403.4314}, 
}
Algorithm 2: EPIC($G_i$, $b$, $\varepsilon_i$)

**Input:** Graph $G_i$, seed set size $b$, and error threshold $\varepsilon_i$.
**Output:** A $b$-size seed set $S_i$ that provides an expected approximation guarantee of at least $\rho_b(1 - \varepsilon_i)$.

1. let $\delta_i \leftarrow 0.01\varepsilon_i/b\cdot n_i$;
2. let $\varepsilon' = (\delta_i - 0.1\cdot n_i)/(b - \delta_i\cdot n_i)$;
3. let $\varepsilon_a = \varepsilon'/(1 - \varepsilon'v)$;
4. let $i_{\text{max}} \leftarrow \lfloor \log_2 2(\log_2 3\varepsilon_a)/\varepsilon^2 \rfloor + 1$ and $a_i \leftarrow \ln 2\cdot \max\{\varepsilon_a, \varepsilon'v\}$;
5. initialize $\theta_0 \leftarrow 1/(\ln 2 + \ln (\varepsilon'\cdot a_i))$;
6. generate two sets $\mathcal{R}_1$ and $\mathcal{R}_2$ of random RR-sets, with $|\mathcal{R}_1| = |\mathcal{R}_2| = \theta_0$;
7. for $i \leftarrow 1$ to $i_{\text{max}}$ do
   8. $(S_i, F^u(S_i^o)) \leftarrow \text{MAXCOVER}(\mathcal{R}_1, b)$;
   9. $F^i(S_i) \leftarrow \left\{ F_{R_2}(S_i) + \frac{2\gamma}{|\mathcal{R}_1|} - \frac{\mu_i}{\ln |\mathcal{R}_2|} \right\}^2 - \frac{\mu_i}{18|\mathcal{R}_2|}$;
10. if $F^i(S_i) / F^u(S_i^o) \geq \rho_b(1 - \varepsilon_i^i)$ or $i = i_{\text{max}}$ then return $S_i$, double the sizes of $\mathcal{R}_1$ and $\mathcal{R}_2$ with new RR-sets;

Algorithm 3: MAXCOVER($\mathcal{R}, b$)

**Input:** A set $\mathcal{R}$ of random RR-sets, and seed set size $b$.
**Output:** A node set $S_i$ and an upper bound $F^u(S_i^o)$ on the fraction of node-sets in $\mathcal{R}$ covered by $S_i$.

1. $S_i \leftarrow \emptyset$;
2. $F^u(S_i^o) \leftarrow \text{MIN}(S_i^o \mid S_i)$ which is computed by (17);
3. for $i \leftarrow 1$ to $b$ do
   4. $u \leftarrow \arg \max_{v \in V_i} (\text{Cov}_{\mathcal{R}}(S_i \cup \{v\}) - \text{Cov}_{\mathcal{R}}(S_i))$;
   5. insert $u$ into $S_i$;
   6. compute $F^u(S_i^o \mid S_i)$ by (17) based on the new $S_i$;
   7. update $F^u(S_i^o) \leftarrow \min\{F^u(S_i^o), F^u(S_i^o \mid S_i)\}$;
8. return $(S_i, F^u(S_i^o))$;

Based on the RR-set sampling method described previously, a simple approach for selecting $S_i$ with a large expected influence spread is to first generate a set $\mathcal{R}$ of RR-sets, and then invoke the MAXCOVER algorithm on $\mathcal{R}$. In particular, MAXCOVER uses a simple greedy approach to identify $S_i \subseteq V_i$ such that $S_i$ overlaps as many RR-sets in $\mathcal{R}$ as possible. Since $F^\mathcal{R}_\mathcal{R}(\cdot)$ is a submodular function for any set $\mathcal{R}$ of RR-sets [3], given any node set $S \subseteq V$ with $|S| \leq b$, we know that

$$F^u(S_i^o \mid S) = F^\mathcal{R}_\mathcal{R}(S) + \sum_{v \in \text{maxMC}(S,b)} (F^\mathcal{R}_\mathcal{R}(S \cup \{v\}) - F^\mathcal{R}_\mathcal{R}(S))$$

(17)

is an upper bound on $F^\mathcal{R}_\mathcal{R}(S_i^o)$, where $S_i^o$ is an optimal seed set in $G_i$ and maxMC($S, b$) is the set of $b$ nodes with the top-$b$ largest marginal coverage in $\mathcal{R}$ with respect to $S$. As a consequence, the smallest one $F^u(S_i^o) = \min_{S_i} \{F^u(S_i^o \mid S_i)\}$ during the greedy procedure ensures that

$$F^u(S_i^o) \geq F^\mathcal{R}_\mathcal{R}(S_i^o).$$

(18)

In addition, according to [33], we also have

$$F^\mathcal{R}_\mathcal{R}(S_i) \geq \rho_b F^u(S_i^o).$$

(19)

where $\rho_b$ is as defined in Algorithm 1. Putting (18) and (19) together yields

$$F^\mathcal{R}_\mathcal{R}(S_i) \geq \rho_b F^u(S_i^o) \geq \rho_b F^\mathcal{R}_\mathcal{R}(S_i^o).$$

(20)

Thus, when $|\mathcal{R}|$ is large, the approximation guarantee of $S_i$ converges to $\rho_b$ according to Equation (20).

To strike a balance between the quality of $S_i$ and the number of RR-sets used to derive $S_i$, EPIC iterates in a careful manner as follows. In each iteration, it maintains two sets of random RR-sets $\mathcal{R}_1$ and $\mathcal{R}_2$ with $|\mathcal{R}_1| = |\mathcal{R}_2|$. It invokes MAXCOVER on $\mathcal{R}_1$ to identify a seed set $S_i$, and then utilizes $\mathcal{R}_2$ to test whether $S_i$ provides a good approximation guarantee. Initially, the cardinalities of $\mathcal{R}_1$ and $\mathcal{R}_2$ are small constants determined by the parameter $\theta_0$ in Line 5 in the first iteration of EPIC. Then, whenever EPIC finds that the quality of the seed set $S_i$ generated in an iteration is not satisfactory, it doubles the sizes of $\mathcal{R}_1$ and $\mathcal{R}_2$. This process repeats until that a qualified solution is identified or the sizes of $\mathcal{R}_1$ and $\mathcal{R}_2$ reach $2^{i_{\text{max}}} - 1$, which exceeds the threshold $2\gamma(2\gamma/3\varepsilon_a)\ln (\ln (\varepsilon'\cdot a_i)))$ (Line 10).

As explained before, one of the main design goals for EPIC is to achieve an expected approximation ratio of $\rho_b(1 - \varepsilon_i)$. EPIC achieves this goal by a series of operations in each iteration, whose implications are briefly explained as follows.

In each iteration, EPIC first applies MAXCOVER on $\mathcal{R}_1$ (Line 8), which returns a seed set $S_i$ and an upper bound $F^u(S_i^o)$ on $F^\mathcal{R}_\mathcal{R}(S_i^o)$, i.e.,

$$F^u(S_i^o) \geq F^\mathcal{R}_\mathcal{R}(S_i^o).$$

(21)

After that, EPIC uses $\mathcal{R}_2$ to estimate the expected spread of $S_i$ (i.e., $E[I_{G_i}(S_i)]$). Observe that $|\mathcal{R}_2|F_{R_2}(S_i)$ is a binomial random variable due to Equation (16). Accordingly, EPIC uses the Chernoff-like martingale concentration bound to set a threshold $F^i(S_i)$ (Line 9) such that

$$E[n_iF_{R_2}(S_i)] \geq n_iF^i(S_i).$$

(22)

should hold with high probability. Intuitively, Equation (22) implies that $n_iF^i(S_i)$ gives a sufficiently accurate lower bound on $E[I_{G_i}(S_i)]$. After that, EPIC checks whether

$$F^i(S_i)/F^u(S_i^o) \geq \rho_b(1 - \varepsilon_i^i)$$

(23)

holds in Line 10. Intuitively, if Equation (23) is true, then we know that $E[n_iF_{R_2}(S_i)]$ is no smaller than $\rho_b(1 - \varepsilon_i^i)n_iF_{R_2}(S_i^o)$ and it suffices to conclude our result by taking the expectation. Specifically, combining Equations (21)–(23) and taking the expectation with respect to the randomness of the algorithm, we can derive a quantitative relationship between $E_{\omega_1}[E[I_{G_i}(S_i)]]$ and $\text{OPT}_b(G_i)$ when a seed set $S_i$ is returned.
where the first inequality is due to the fact that (22) holds with high probability and $\delta_i \rho_b (1 - \varepsilon'_i) n_i$ is used to offset the failed scenario, and the equality is due to the martingale stopping theorem [25] (see details in Sect. 4.3). This proves the $\rho_b (1 - \varepsilon'_i)$ expected approximation ratio of $\mathbb{E}_a[\mathbb{E}[I_{G_i}(S_i)]]$ as $\varepsilon_i = \varepsilon'_i + (1 - \varepsilon'_i)\delta_i n_i / b$.

It is easy to see that the better expected approximation guarantee of EPIC is better than those of vanilla IM algorithms, and thus instantiating ADAPTRGREEDY using EPIC can lead to performance improvement for adaptive IM. Note that EPIC does not provide the worst-case approximation guarantee with high probability, as against the state-of-the-art IM algorithms. The reason behind is that $n_i F_u(S'_i)$ is likely to be smaller than $\text{OPT}_b(G_i)$ though $n_i F_u(S'_i)$ is an upper bound on $n_i F_{\mathcal{R}_i}(S'_i)$ as shown in (21).

### 4.3 Theoretical analysis of EPIC

Based on the discussions in Sect. 4.2, we show the details of theoretical analysis of EPIC. We prove our main results for the expected approximation guarantee and the time complexity of EPIC as follows.

**Expected Approximation Guarantee.** We establish the expected approximation guarantee of EPIC in the following theorem.

**Theorem 4** For any $G_i$, EPIC returns a seed set $S_i$ satisfying

$$
\mathbb{E}_a[\mathbb{E}[I_{G_i}(S_i)]] \geq \rho_b (1 - \varepsilon_i) \text{OPT}_b(G_i). \tag{24}
$$

To prove Theorem 4, we first prove the following lemma.

**Lemma 7** Let $\varepsilon_a \in (0, 1)$, $\delta_a \in (0, 1)$, and

$$
\theta_{\text{max}} = \left( \frac{2 + 2\varepsilon_a / 3}{\varepsilon_a^2 b} \right) \left( \frac{1}{\delta_a} + \ln \left( \frac{n}{b} \right) \right). \tag{25}
$$

If a set of random RR-sets $\mathcal{R}$ are generated such that $|\mathcal{R}| \geq \theta_{\text{max}}$, then with probability at least $1 - \delta_a$, the greedy algorithm returns a solution $S_i$ satisfying

$$
\mathbb{E}[I_{G_i}(S_i)] \geq \frac{\rho_b}{1 + \varepsilon_a} n_i F_{\mathcal{R}_i}(S'_i). \tag{26}
$$

**Proof** (Lemma 7) According to Chernoff-like martingale concentration bound [37], for any $S$ that is independent of $\mathcal{R}$, we have

$$
\Pr \left[ n_i F_{\mathcal{R}}(S) > (1 + \varepsilon_a) \mathbb{E}[n_i F_{\mathcal{R}}(S)] \right] = \exp \left( -\frac{\varepsilon_a^2}{2 + 2\varepsilon_a / 3} \frac{|\mathcal{R}| - \mathbb{E}[n_i F_{\mathcal{R}}(S)]}{n} \right) \leq \delta_a / (n / b).
$$

By the union bound, $S_i$ returned by the greedy algorithm satisfies

$$
\Pr \left[ n_i F_{\mathcal{R}}(S_i) > (1 + \varepsilon_a) \mathbb{E}[n_i F_{\mathcal{R}}(S_i)] \right] \leq \delta_a. \tag{27}
$$

On the other hand, due to the submodularity of $F_{\mathcal{R}}(\cdot)$, we have

$$
F_{\mathcal{R}}(S_i) \geq \rho_b F_{\mathcal{R}}(S'_i). \tag{28}
$$

Combining (27) and (28) completes the proof.

Next, we use the following martingale stopping theorem [25] to prove Theorem 4.

**Definition 6** (Stopping Time [25]) A nonnegative, integer-valued random variable $T$ is a stopping time for the sequence $\{Z_n, n \geq 0\}$ if the event $T = n$ depends only on the value of the random variables $Z_0, Z_1, \ldots, Z_n$.

**Lemma 8** (Martingale Stopping Theorem [25]) If $Z_0, Z_1, \ldots$ is a martingale with respect to $Y_1, Y_2, \ldots$ and if $T$ is a stopping time for $Y_1, Y_2, \ldots$, then

$$
\mathbb{E}[Z_T] = \mathbb{E}[Z_0] \tag{29}
$$

whenever one of the following holds:

- the $Z_i$ are bounded, so there is a constant $c$ such that, for all $i$, $|Z_i| \leq c$;
- $T$ is bounded;
- $\mathbb{E}[T] < \infty$, and there is a constant $c$ such that $\mathbb{E}[|Z_{i+1} - Z_i| \mid Y_1, \ldots, Y_i] < c$.

**Proof** (Theorem 4) Let $\mathcal{E}_1$ and $\mathcal{E}_2$ denote the following events:

$$
\mathcal{E}_1(S_i) : \mathbb{E}[I_{G_i}(S_i)] \geq \rho_b (1 - \varepsilon'_i) n_i F_{\mathcal{R}_i}(S'_i),
$$

$$
\mathcal{E}_2(S_i) : \mathbb{E}[I_{G_i}(S_i)] \geq n_i F^i(S_i).
$$

Let $T$ be the stopping time (i.e., the iteration in which EPIC returns $S_i$), which is bounded by $i_{\text{max}}$. Let $\varepsilon_a = \varepsilon'_i / (1 - \varepsilon'_i)$ and $\delta_a = \delta_i / 2$ for $\theta_{\text{max}}$ defined in (25). When $T = i_{\text{max}}$, it
is easy to verify that \(|\mathcal{R}_1| = 2^{i_{\text{max}} - 1}\theta_0 \geq \theta_{\text{max}}\). Hence, by Lemma 7, we have

\[
\mathbb{P}[(T = i_{\text{max}}) \wedge \neg \mathcal{C}_1(S_t)] \leq \delta_i / 2. \tag{30}
\]

On the other hand, when \(T = t < i_{\text{max}}\), let \(S_t\) be the set of possible node sets selected by EPIC (but not necessarily returned), where each \(S \in S_t\) has a probability \(\mathbb{P}[S]\) such that \(\sum_{S \in S_t} \mathbb{P}[S] = 1\). Then, we have

\[
\mathbb{P}[(T = t) \wedge \neg \mathcal{C}_1(S_t)] \\
\leq \sum_{S \in S_t} \mathbb{P}[(T = t) \wedge \neg \mathcal{C}_2(S)] \cdot \mathbb{P}[S_t = S] \\
\leq \sum_{S \in S_t} \delta_i / (2i_{\text{max}}) \cdot \mathbb{P}[S_t = S] \\
\leq \delta_i / (2i_{\text{max}}).
\]

where the first inequality is because if \(\mathcal{C}_2\) happens then \(\mathcal{C}_1\) must also happen, the second inequality is by the fact that only a subset of the node sets in \(S_t\) are returned, and the third inequality is obtained from [33] for any node set \(S\) that is independent of \(\mathcal{R}_2\). As a consequence, by a union bound,

\[
\mathbb{P}\left[\bigvee_{i=1}^{i_{\text{max}} - 1} (T = t) \wedge \neg \mathcal{C}_1(S_t)\right] \leq \delta_i / 2. \tag{31}
\]

Combining Equations (30) and (31) shows that the event \(\mathcal{C}_1(S_t)\) does not happen with probability at most \(\delta_i\) no matter when the algorithm stops. Therefore, EPIC returns a random solution \(S_t\) satisfying \(\mathbb{E}[I_{\mathcal{C}_1}(S_t)] \geq \rho_b(1 - \varepsilon'_i)n_i F_{\mathcal{R}_1}(S'_t)\) with at least \(1 - \delta_i\) probability. Thus, adding an additive factor of \(\delta_i \rho_b(1 - \varepsilon'_i)n_i\) ensures that

\[
\mathbb{E}_\omega[\mathbb{E}[I_{\mathcal{C}_1}(S_t)] + \delta_i \rho_b(1 - \varepsilon'_i)n_i] \\
\geq \rho_b(1 - \varepsilon'_i)\mathbb{E}_\omega[n_i F_{\mathcal{R}_1}(S'_t)]. \tag{32}
\]

Subsequently, the main challenge lies in how we connect \(n_i F_{\mathcal{R}_1}(S'_t)\) with \(\text{OPT}_b(G_i)\). Note that \(F_{\mathcal{R}_1}(S'_t)\) is also a random variable with respect to \(\mathcal{R}_1\). At the first glance, it seems that this analysis is difficult as the stopping time is a random variable. However, fortunately, by utilizing the martingale stopping theorem [25], we can bridge the gap between \(n_i F_{\mathcal{R}_1}(S'_t)\) and \(\text{OPT}_b(G_i)\) as follows.

Note that \(T\) is bounded within \(i_{\text{max}}\) so that it satisfies the condition of martingale stopping theorem as shown in Lemma 8. Thus, we have

\[
\mathbb{E}_\omega[n_i F_{\mathcal{R}_1}(S'_t)] - \text{OPT}_b(G_i) = 0. \tag{33}
\]

Combining Equations (32) and (33) yields:

\[
\mathbb{E}_\omega[\mathbb{E}[I_{\mathcal{C}_1}(S_t)] \geq \rho_b(1 - \varepsilon'_i)(1 - \delta_i n_i / b)\text{OPT}_b(G_i). \tag{34}
\]

By replacing \(\varepsilon'_i = (b - \delta_in_i)/(b - \delta_1n_i)\), we can immediately acquire Equation (24), by which we complete the proof of Theorem 4. \(\square\)

**Time complexity.** The expected time complexity of EPIC is given in the following theorem.

**Theorem 5** For any \(G_i\), the expected time complexity of EPIC is \(O((b \log n_i + \log 1 / \varepsilon'_i)(m_i + n_i)/\varepsilon'_i^2)\), where \(n_i\) and \(m_i\) are the number of nodes and edges of \(G_i\), respectively.

**Proof** (Theorem 5) When we set the parameters \((\varepsilon'_i, \delta_i)\) for \(\text{OPIM-C}\) [33], the expected time complexity of \(\text{OPIM-C}\) is

\[
O\left(\frac{m_i + n_i}{\varepsilon'_i^2} \left( b \log n_i + \log \frac{1}{\varepsilon'_i} \right) \right) \\
= O\left(\frac{m_i + n_i}{\varepsilon'_i^2} \left( b \log n_i + \log \frac{1}{\varepsilon'_i} \right) \right).
\]

On the other hand, for any given \(\mathcal{R}_1\) and \(\mathcal{R}_2\), if \(\text{OPIM-C}\) stops, then EPIC must also stop. This implies that EPIC always finishes earlier than \(\text{OPIM-C}\), which completes the proof. \(\square\)

### 4.4 **ADAPTGREEDY instantiated with EPIC**

In the following, we derive the approximation guarantees and time complexity of ADAPTGREEDY instantiated using EPIC.

Theorem 4 indicates that ADAPTGREEDY instantiated using EPIC with parameter \(\varepsilon_i\) achieves an expected approximation guarantee of at least \(\rho_b(1 - \varepsilon_i)\) in the \(i\)-th batch of seed selection. Immediately following by Theorem 3 and Theorem 5, we have the following theorem.

**Theorem 6** Suppose that we instantiate ADAPTGREEDY using EPIC with parameter \(\varepsilon_i\) for the \(i\)-th batch of seed selection, then ADAPTGREEDY achieves the expected approximation ratio of \(1 - \varepsilon^{\rho_b(c^\varepsilon - 1)}\) where \(\varepsilon = \frac{1}{\varepsilon'_i} \sum_{i=1}^{i_{\text{max}}} \varepsilon_i\), and takes an expected time complexity of \(O\left(\sum_{i=1}^{i_{\text{max}}} (b \log n + \log 1 / \varepsilon'_i) (m + n) / \varepsilon'_i^2\right)\).

To achieve the expected approximation ratio of \(1 - \varepsilon^{\rho_b(c^\varepsilon - 1)}\), instantiating ADAPTGREEDY using EPIC takes shorter running time compared with that of using the naive expected approximation guarantee of the existing IM algorithms. As discussed in Sect. 4.2, the intuition behind is that EPIC avoids the additional estimation error on \(\text{OPT}_b(G_i)\) which is considered by all the existing IM algorithms.

In addition, Theorem 3 indicates that ADAPTGREEDY instantiated using EPIC with parameter \(\varepsilon_i\) achieves the worst-case approximation ratio of \(1 - \varepsilon^{\rho_b(c^\varepsilon - 1)}\) with a probability
of at least $1 - \delta$, where $\epsilon' = \frac{1}{r} \sum_{i=1}^{r} \epsilon_i + \sqrt{\frac{1}{2r} \cdot \ln(1/\delta)}$. Therefore, to achieve a predefined worst-case approximation ratio of $1 - e^{\rho_b(\epsilon - 1)}$ with a probability of at least $1 - \delta$, we may decrease the parameter $\epsilon_i$ in EPIC by an additive factor of $\sqrt{\frac{1}{2r} \cdot \ln(1/\delta)}$ for every $i$.

**Theorem 7** Suppose that we instantiate ADAPTMEDG using EPIC with the parameters $\epsilon' = \epsilon_i - \sqrt{\frac{1}{2r} \cdot \ln(1/\delta)}$ in each batch where $\delta \in (0, 1)$. Then, ADAPTMEDG achieves the approximation ratio $1 - e^{\rho_b(\epsilon - 1)}$ with a probability of at least $1 - \delta$ where $\epsilon = \frac{1}{r} \sum_{i=1}^{r} \epsilon_i$, and takes an expected time complexity of $O(\sum_{i=1}^{r} (b \log n + \log \frac{b}{r})(m + n)/\epsilon_i^2)$.

Note that Theorem 7 requires that $\epsilon' = \epsilon_i - \sqrt{\frac{1}{2r} \cdot \ln(1/\delta)} > 0$. $r$ is sufficiently large, i.e., $r > \frac{\ln(1/\delta)}{2\epsilon}$, there is a valid instantiation of ADAPTMEDG to achieve a predefined worst-case approximation guarantee of $1 - e^{\rho_b(\epsilon - 1)}$ with probability at least $1 - \delta$.

### 5 Misclaims in previous work [16,31]

In this section, we revisit two of the latest work proposed to address the adaptive IM problem, i.e., our preliminary work [16] and Sun et al.’s work [31]. We aim to discuss potential issues and clarify some common misunderstandings towards this problem. Specifically, their algorithms are claimed to return a worst-case approximation guarantee with high probability. However, there exist potential theoretical issues in the analysis of the failure probability, which is elaborated as follows.

In [16] (Section 4.1), it is claimed that the overall failure probability of the $i$-th batch satisfies

$$
\Pr \left[ \epsilon_i < \sum_{G_1, \ldots, G_i} (\xi_i \cdot \Pr[\xi_i, G_1, \ldots, G_{i-1}]) \right] \leq \delta_i.
$$

Then, the failure probability of all $r$ batches is bounded by a union bound of $\sum_{i=1}^{r} \delta_i$.

Similarly, in [31] (Theorem 5.5) it is claimed that the proposed algorithm ADAIMM achieves the worst-case approximation with $1 - \frac{1}{n^l}$ probability where $l$ is a constant. They first prove that the seed set $S$ selected by ADAIMM returns an approximation with at least $1 - 1/(n^l \cdot r)$ probability for each batch. Sun et al. [31] thus claim that ADAIMM achieves the approximation ratio with at least $1 - 1/n^l$ probability by union bound.

The theoretical guarantees of these two papers are based on Theorem A.10 in [14]. Through a careful examination of the proof of Theorem A.10 in [14], we find that the essence is to bound the overall approximation guarantee for each batch, i.e., $X_i(\omega) \geq \rho_b(1 - \epsilon_i)$, where $X_i(\omega)$ represents the overall random approximation for the $i$-th batch of seed selection over all realizations, i.e.,

$$
X_i(\omega) := \frac{\mathbb{E}_{G_i} [\rho_b(1 - \xi_i, G_i) \cdot \OPT_b(G_i)]}{\mathbb{E}_{G_i} [\OPT_b(G_i)]}.
$$

Since $X_i(\omega)$ is a random variable that is likely to be smaller than $\rho_b(1 - \epsilon_i)$, these two papers [16,31] attempt to bound the probability of $X_i(\omega) < \rho_b(1 - \epsilon_i)$ as

$$
\Pr[\epsilon_i < \rho_b(1 - \epsilon_i)] \leq \delta_i.
$$

However, as long as there exists one realization such that the seed set $S_i$ returned in the $i$-th batch does not meet the approximation of $\rho_b(1 - \epsilon_i)$, i.e., $\xi_i(\omega, G_i) > \epsilon_i$, it is possible that $X_i(\omega) < \rho_b(1 - \epsilon_i)$. On the other hand, there are exponential number $O(2^m)$ of realizations, where $m$ is the number of edges in $G$. Thus, although it holds that $\Pr[\xi_i(\omega, G_i) > \epsilon_i] \leq \delta_i$ under a given $G_i$, the probability of $X_i(\omega) < \rho_b(1 - \epsilon_i)$ can be as large as $O(2^m \cdot \delta_i)$ by the union bound. Therefore, it is intricate to bound $X_i(\omega)$, which indicates that their claims on failure probability do not hold. In other words, Theorem 4 in [16] and Theorem 5.5 in [31] are invalid.

In this paper, we rectify the theoretical analysis of the worst-case approximation guarantee utilizing Azuma-Hoeffding inequality [25]. In particular, instead of bounding the probability of each individual $X_i(\omega) < \rho_b(1 - \epsilon_i)$, we directly bound the probability of $X(\omega) < \rho_b(1 - \epsilon)$, where $X(\omega) = \frac{1}{r} \sum_{i=1}^{r} X_i(\omega)$ and $\epsilon = \frac{1}{r} \sum_{i=1}^{r} \epsilon_i$, as $X(\omega)$ should be concentrated to its expectation $\mathbb{E}_\omega[X(\omega)]$ when $r$ is sizable and $\mathbb{E}_\omega[X_i(\omega)] \geq \rho_b(1 - \epsilon_i)$ can be achieved by various non-adaptive IM algorithms.

### 6 Related work

#### 6.1 Comparison with the preliminary version

Compared with our preliminary work [16], the current paper includes two major new contributions as follows.

First, we propose a randomized greedy policy that can provide strong theoretical guarantees for the general adaptive stochastic maximization problem, which may be of independent interest. This new solution can be adopted in many other settings apart from adaptive IM, e.g., active learning [11], active inspection [17], optimal information gathering [10], which are special cases of adaptive stochastic maximization. In particular, our proposed policy imposes far fewer constraints than Golovin and Krause’s policy [14]. In fact, in some applications (e.g., adaptive IM), the requirement of Golovin and Krause’s policy [14] is too stringent to construct such a policy whereas our proposed policy is easy to obtain. Moreover, we show that our policy can achieve a worst-case approximation guarantee with high probability, which uncovers some
potential gaps in two recent studies [16,31] and shed light on the future work of the adaptive IM problem.

Second, we improve the efficiency of algorithm EPIC (Sect. 4). In [16], EPIC is designed based on an idea similar to that of the SSA algorithm in [26]. However, SSA is rather inefficient when the input error parameter $\varepsilon$ is small, as verified in [33]. Therefore, we redesigned EPIC based on the state-of-the-art method OPIM- C [33], which is far more efficient than SSA. Moreover, we optimize the estimation of the upper bound of OPT in EPIC based on martingale stopping theorem [25], which boosts the performance of ADAPTGREEDY noticeably.

6.2 Non-adaptive influence maximization

The IM problem under the non-adaptive setting has been extensively studied. The seminal work of Kempe et al. [20] shows that there is a $1 - 1/e - \varepsilon$ approximation guarantee for the non-adaptive IM problem, and it proposes a monte carlo simulation algorithm to achieve this approximation ratio with high time complexity. After that, a lot of studies have appeared to improve Kempe et al.’s work in terms of time efficiency, especially for some applications [23,24] that require efficient algorithms to identify the top-$k$ influential set in large graphs. Among these works, Borgs et al. [3] propose the RR-set sampling method for influence spread estimation, and several later studies [26,33,36,37] use this method to find more efficient algorithms for the IM problem. However, all these studies concentrate on the non-adaptive IM problem, and hence their approximation guarantees do not hold for the adaptive IM problem.

6.3 Adaptive influence maximization

Compared with the studies on non-adaptive IM, the studies on adaptive IM are relatively few. Golovin et al. [14] derive a $(1 - 1/e)$-approximation ratio under the case that only one seed node can be selected in each batch. The feedback model they consider is the same as the one described in this paper, which they call the full-adoption feedback model. In their arXiv version, they also mention another feedback model called myopic feedback model, where the feedback of a selected seed node only includes the directed neighbors activated by the seed, but does not include further activated nodes in the cascade process. They show that under the IC model full-adoption feedback is adaptive submodular but myopic feedback is not adaptive submodular. In addition, Yuan and Tang [40] propose a generalized feedback model, called partial feedback model, under which the objective is not adaptive submodular either. Chen et al. [9], Tang et al. [35], Vaswani and Lakshmanan [38] study adaptive seed selection under the case that more than one seed nodes can be selected in each batch. Nevertheless, Chen et al. [9] and Tang et al. [35] aim to minimize the cost of the selected seeds under the constraint that the influence spread is larger than a given threshold, which is a different goal from ours. Vaswani and Lakshmanan [38] derive an approximation guarantee $1 - \exp\left(-\frac{1}{\eta}\right)$ for certain $\eta > 1$. Unfortunately, none of the studies listed above provide a practical algorithm to achieve the claimed approximation ratios. More specifically, Golovin et al. [14] and Chen et al. [9] assume that the expected influence spread can be exactly computed in polynomial time (which is not true due to [6]), while Vaswani and Lakshmanan [38] did not provide a method to bound the key parameter $\eta$ appearing in their approximation ratio. Sun et al. [31] study the Multi-Round Influence Maximization (MRIM) problem under the multi-round triggering model, where influence propagates in multiple rounds independently from possibly different seed sets. In our adaptive IM problem, we consider a natural diffusion model that the realization of influence propagation is identical for all batches/rounds. Meanwhile, as we discussed, our analyses of approximation guarantees uncover some potential gaps in [31].

More recently, there are a few studies on the adaptivity gap, the ratio between the optimal adaptive solution versus the optimal non-adaptive solution, in the context of adaptive influence maximization. Peng and Chen [29] show a constant adaptivity gap for adaptive influence maximization under the IC model with myopic feedback, and using this result to further show that the adaptive greedy algorithm achieves a constant approximation even though the model is not adaptive submodular. They also show in another paper [4] the constant upper and lower bounds for the adaptivity gap in the IC model with full-adoption feedback for several special classes of graphs, but the adaptivity gap for the general graphs remains open. Chen et al. [8] define the greedy adaptivity gap as the ratio between the adaptive greedy solution versus the non-adaptive greedy solution, and provide upper/lower bounds for the greedy adaptivity gap under certain influence propagation models. These studies on the adaptivity gap demonstrate the power and limitation of adaptivity in influence maximization and are complementary to our study on efficient algorithms for adaptive influence maximization.

We also note that Seeman et al. [30], Horel et al. [18] and Badanidiyuru et al. [2] consider an influence maximization problem called “adaptive seeding”, but with totally different implications from ours. More specifically, they assume that the seed nodes can be selected in two stages. In the first stage, a set $S$ can be selected from a given node set $S' \subseteq V$. In the second stage, another seed set $S^{+}$ can be selected from the influenced neighboring nodes of $S$. The goal of their problem is to maximize the expected influence spread of $S^{+}$, under the constraint that the total number of nodes in $S \cup S^{+}$ is no more than $k$. However, both the problem model and the optimization goal of these studies are very different.
In Sect. 4.1. In addition, a worst-case approximation ratio of $1 - \frac{1}{\epsilon}$ was pointed out in Theorem 7. EptAIM-N is a naive instantiation of ADAPTGREEDY instantiated using the existing non-adaptive algorithm OPIM-C [33], directly, as introduced in Sect. 4.1. In addition, EptAIM-N fixes the issues of ADAPTIM-1 [16] so that it provides the correct expected approximation ratio of $1 - \frac{\rho_b}{\epsilon}$. By including EptAIM-N, we could evaluate the performance improvement of EptAIM against EptAIM-N. FixAIM is a variant of EptAIM that uses a fixed number of samples for each batch of seed selection. Note that FixAIM is a heuristic algorithm, which does not provide any theoretical guarantees. The purpose of FixAIM is to provide some insights on the effect of sample size on the performance of adaptive algorithms.

We also test two state-of-the-art non-adaptive IM algorithms (i.e., IMM [37] and D-SSA [26]) in our experiments. The purpose of using D-SSA and IMM in our experiments is to measure the influence spread increase achieved by the adaptive IM algorithms compared with the non-adaptive IM algorithms.

### Parameter settings

We use the popular independent cascade (IC) model [20] in our experiments. Following a large body of existing work on influence maximization [20, 26, 33, 36, 37], we set the propagation probability of each edge $(u, v)$ to $\frac{1}{d_u(v)}$, where $d_u(v)$ is the in-degree of node $v$. We set $\epsilon = 0.5$ for the three adaptive algorithms and two non-adaptive algorithms for fair comparison and approximation errors $\epsilon_1 = \cdots = \epsilon_r = \epsilon$ for the three adaptive algorithms. Meanwhile, we set the failure probability of $\delta = 1/n$ for WstAIM, IMM, and D-SSA. For FixAIM, we generate 10K RR-sets for each batch of seed selection.

Recall that we need to select $k$ nodes in $r$ batches in adaptive IM, where $b = k/r$ nodes are selected in each batch. To see how the performance of our algorithms is affected by input parameters $k, b$, and $r$, we set these parameters according to the $b$-setting and $k$-setting explained as follows. Under the $b$-setting, we fix $k = 500$ and vary $b$ such that $b \in \{1, 2, 4, 5, 10, 500\}$. Under the $k$-setting, we fix $r = 50$ and vary $k$ such that $k \in \{50, 100, 200, \ldots, 500\}$.

### 7.2 Comparison of influence spread

In this section, we study the influence spread for all tested algorithms, as shown in Fig. 3 and 4. In order to gain a comprehensive understanding about the efficacy of the tested algorithms, we measure their influence spreads achieved by varying the number of seed nodes $k$ and the batch size $b$.

Fig. 3 reports the influence spread obtained with 500 seed nodes selected through different numbers of batches on the four datasets. In general, the spreads acquired by EptAIM, WstAIM, and EptAIM-N are comparable to each other but notably larger than the spreads of the baselines, including the heuristic adaptive algorithm, i.e., FixAIM, and two non-adaptive algorithms, i.e., IMM and D-SSA. In particular, WstAIM, IMM and D-SSA achieve the worst-case approximation guarantee, while WstAIM obtains around 12% and 60% more spread than IMM and D-SSA do in average, respectively. On the one hand, this can be explained by the advantage of adaptivity over non-adaptivity that adaptive algorithms could make smarter decisions based on the

### 7 Performance evaluation

In this section, we evaluate the performance of our proposed approach with extensive experiments. The goal of our experiments is to measure the efficiency and effectiveness of ADAPTGREEDY using real social networks. All of our experiments are conducted on a Linux machine with an Intel Xeon 2.6GHz CPU and 64GB RAM.

#### 7.1 Experimental setting

**Datasets.** We use five real datasets in our experiments, i.e., NetHEPT, Epinions, DBLP, LiveJournal, and Orkut as summarized by Table 2. NetHEPT is obtained from [5], representing the academic collaboration networks in “High Energy Physics-Theory” area. The rest four datasets are available from [21]. Among them, Orkut contains millions of nodes and edges. We randomly generate 20 realizations for each dataset, and then report the average performance for each algorithm on those 20 realizations.

**Algorithms.** We evaluate four adaptive algorithms, i.e., EptAIM, WstAIM, EptAIM-N, and FixAIM and two state-of-the-art non-adaptive algorithms, i.e., IMM [37] and D-SSA [26]. EptAIM is the algorithm we instantiate ADAPTGREEDY with EPIC to achieve an expected approximation ratio of $1 - e^{\rho_b(\epsilon - 1)}$ where $\rho_b = 1 - (1 - 1/b)^b$ and $b$ is the batch size. WstAIM is the same implementation as EptAIM but with well-calibrated parameters to acquire a worst-case approximation ratio of $1 - e^{\rho_b(\epsilon - 1)}$ with high probability. Recall that obtaining the worst-case approximation needs a more demanding requirement than the expected approximation, pointed out in Theorem 7. EptAIM-N is a naive instantiation of ADAPTGREEDY instantiated using the existing non-adaptive algorithm OPIM-C [33] directly, as introduced in Sect. 4.1. In addition, EptAIM-N fixes the issues of ADAPTIM-1 [16] so that it provides the correct expected approximation ratio of $1 - e^{\rho_b(\epsilon - 1)}$. By including EptAIM-N, we could evaluate the performance improvement of EptAIM against EptAIM-N. FixAIM is a variant of EptAIM that uses a fixed number of samples for each batch of seed selection. Note that FixAIM is a heuristic algorithm, which does not provide any theoretical guarantees. The purpose of FixAIM is to provide some insights on the effect of sample size on the performance of adaptive algorithms.

We also test two state-of-the-art non-adaptive IM algorithms (i.e., IMM [37] and D-SSA [26]) in our experiments. The purpose of using D-SSA and IMM in our experiments is to measure the influence spread increase achieved by the adaptive IM algorithms compared with the non-adaptive IM algorithms.

**Parameter settings.** We use the popular independent cascade (IC) model [20] in our experiments. Following a large body of existing work on influence maximization [20, 26, 33, 36, 37], we set the propagation probability of each edge $(u, v)$ to $\frac{1}{d_u(v)}$, where $d_u(v)$ is the in-degree of node $v$. We set $\epsilon = 0.5$ for the three adaptive algorithms and two non-adaptive algorithms for fair comparison and approximation errors $\epsilon_1 = \cdots = \epsilon_r = \epsilon$ for the three adaptive algorithms. Meanwhile, we set the failure probability of $\delta = 1/n$ for WstAIM, IMM, and D-SSA. For FixAIM, we generate 10K RR-sets for each batch of seed selection.

Recall that we need to select $k$ nodes in $r$ batches in adaptive IM, where $b = k/r$ nodes are selected in each batch. To see how the performance of our algorithms is affected by input parameters $k, b$, and $r$, we set these parameters according to the $b$-setting and $k$-setting explained as follows. Under the $b$-setting, we fix $k = 500$ and vary $b$ such that $b \in \{1, 2, 4, 5, 10, 500\}$. Under the $k$-setting, we fix $r = 50$ and vary $k$ such that $k \in \{50, 100, 200, \ldots, 500\}$.

| Dataset    | $n$     | $m$     | Type     | Avg. deg |
|------------|---------|---------|----------|----------|
| NetHEPT    | 15.2K   | 31.4K   | Undirected | 4.18     |
| Epinions   | 132K    | 841K    | Directed  | 13.4     |
| DBLP       | 655K    | 1.99M   | Undirected | 6.08     |
| LiveJournal| 4.85M   | 69.0M   | Directed  | 28.5     |
| Orkut      | 3.07M   | 117M    | Undirected | 76.2     |
feedback from previous batches. On the other hand, the considerable discrepancy on the spread of D-SSA exposes that D-SSA sacrifices its effectiveness badly for the sake of high efficiency (referring to its running time, as shown in Fig. 6 and 7). As with FixAIM, it achieves the smallest spreads among the four adaptive algorithms, with around 10% less in average. In particular, FixAIM obtains even smaller spreads than the non-adaptive algorithm IMM on the three largest datasets, as shown in Fig. 3. This fact suggests that 10K samples are insufficient to provide good performance.

Fig. 4 shows the results of influence spreads with various number of seed nodes. We observe that (i) the spreads grow with the number of seed nodes $k$ as expected, (ii) our three adaptive algorithms achieve similar amount of spreads and outperform the heuristic adaptive algorithm FixAIM under the same $k$ and $b$ setting, which is consistent with the results in Fig. 3, and (iii) the percentage increase of spreads obtained by the adaptive algorithms over the spreads of IMM is around 10% in average. This spread improvement is quite promising considering the large number of users in social networks. Meanwhile, it further confirms the superiority of adaptive algorithms on influence maximization.

Fig. 5 reports the average spread of our proposed EptAIM algorithm under different number of realizations, including 10, 20, 40, 50, 100, on the NetHEPT dataset. As shown, the average spreads of EptAIM in various numbers of realizations are well-converged, especially under the $k$-setting. These results support the reliability of the results obtained through 20 random realizations.

7.3 Comparison of running time

In this section, we investigate the efficiency of all tested algorithms under various seed node numbers $k$ and batch sizes $b$.

The settings of Fig. 6 and 7 follow the settings of Fig. 3 and 4, respectively. In particular, Fig. 6 reports the running time with $k = 500$ and various $b$ values under the four datasets. We observe that among the four adaptive algorithms, FixAIM surpasses the other three adaptive algorithms significantly as expected, and can even beat the non-adaptive algorithm IMM in some circumstances. This is because FixAIM generates a small number of samples (i.e., 10K) for each batch. In addition, EptAIM dominates the other two adaptive algorithms on all datasets with a non-negligible advantage. Specifically, the performance gap between EptAIM and WstAIM tends to enlarge along the increase of the batch size $b$. This expanding gap is due to that (i) to maintain the same approximation ratio, WstAIM needs to compensate for an extra factor $\sqrt{1/2(2r) \cdot \ln(1/\delta)}$ on approximation error for
each batch, as explained in Theorem 7, and (ii) when the seed number \( k \) is fixed, this compensation factor gets larger since the batch number \( r = k/b \) gets smaller. Note that E\(_{PTAIM}\) - N runs slower than E\(_{PTAIM}\) for all cases on the four datasets. When batch size \( b = 1 \), the efficiency gap can be up to 3 times, which demonstrates the speed improvement of our optimization in E\(_{PTAIM}\). One interesting observation is that W\(_{STAIM}\) has a slightly edge over E\(_{PTAIM}\) - N when the batch size \( b \leq 5 \).

Another noticeable observation is that the running time increases along the decrease of batch size \( b \). There are two main reasons. First, when the batch size \( b \) becomes smaller, the marginal spread drops significantly. To maintain the same approximation, more samples are generated, which incurs considerable overhead. Second, when the number of seeds \( k \) is fixed, larger \( b \) value means smaller value of \( r \). As mentioned, RR-sets are regenerated for each batch, and thus, a fewer number of batches leads to less sampling overhead.

Fig. 7 plots the running time when both seed number \( k \) and batch size \( b \) vary while the batch number is fixed to \( r = 50 \). Again, FIXAIM runs faster than the other three adaptive algorithms, and IMM for some cases. We also observe that the running time of FIXAIM remains approximately constant, since its running time is roughly linear in the number of rounds \( r \) which is a constant, i.e., \( r = 50 \). In addition, we can see that E\(_{PTAIM}\) outperforms the other two adaptive algorithms with around 1.5–3 times speedup. Second, under this setting, the running time of W\(_{STAIM}\) is comparable with that of E\(_{PTAIM}\) - N. Observe that the running time of the adaptive algorithms does not fluctuate as much as that in Fig. 6 when the seed size \( k \) changes.

This observation demonstrates that adaptive algorithms are more sensitive to the value of batch \( r \) than the seed size \( k \).

Note that the two non-adaptive algorithms dominate the three adaptive algorithms in efficiency, as expected. This is because non-adaptive algorithms can be seen as special adaptive algorithms with just running in one batch, which avoids enormous sampling time.

### 8 Comparison of memory consumption

Fig. 8 presents the memory consumptions of the tested algorithms. As shown, FIXAIM and two non-adaptive algorithms, i.e., IMM and D- SSA, use the least memory, which remains nearly constant along with the seed size \( k \). The other three adaptive algorithms, i.e., E\(_{PTAIM}\), E\(_{PTAIM}\) - N, and W\(_{STAIM}\), consume relatively larger memory, especially for \( k = 50 \), in which case the batch size \( b = 1 \). Among them, W\(_{STAIM}\) needs the most memory. Observe that the memory consumptions of the adaptive algorithms approach to those of the non-adaptive algorithms when the seed size \( k \) increases. To explain, the batch size \( b \) increases along with the seed size \( k \), which indicates that less samples would be generated for each batch of seed selection for OPIM- C [33] based adaptive algorithms. Meanwhile, adaptive algorithms would remove all samples generated in previous batches, which could save memory significantly. Note that all memory consumptions are close on the Orkut dataset, since the memory taken up to store the graph itself dominates the whole memory usage.
9 Conclusion and future work

We have studied the adaptive influence maximization (IM) problem, where the seed nodes can be selected in multiple batches to maximize their influence spread. We have proposed the first practical algorithm to address the adaptive IM problem that achieves both time efficiency and provable approximation guarantee. Specifically, our approach is based on a novel ADAPTGREEDY framework instantiated by a new non-adaptive IM algorithm EPIC, which has a provable expected approximation guarantee for non-adaptive IM. Meanwhile, we have clarified some existing misunderstandings in two recent work towards the adaptive IM problem and laid solid foundations for further study. Our solution to the adaptive influence maximization is based on our general solution to the adaptive stochastic maximization problem with a randomized approximation algorithm at every adaptive greedy step, and this general solution could be useful to many other settings besides adaptive influence maximization. We have also conducted extensive experiments using real social networks to evaluate the performance of our algorithms, and the experimental results strongly corroborate the superiorities and effectiveness of our approach.

For future work, we aim to devise new algorithms that could reuse the samples generated in previous batches to further boost the efficiency. Specifically, for unbiased spread estimation in each batch of seed selection, our current algorithms generate sufficient number of RR-sets by abandoning all samples generated in previous batches. The reason behind is that reusing the “old” samples generated in previous batches could incur bias for spread estimation, which will affect seed selection. To tackle this issue, we aim to develop new techniques to fix or bound the bias by sample reuse, which is expected to boost the efficiency remarkably.

References

1. Arora, A., Galhotra, S., Ranu, S.: Debunking the myths of influence maximization: an in-depth benchmarking study. In: Proceedings of ACM SIGMOD, pp. 651–666 (2017)
2. Badanidiyuru A, Papadimitriou, C., Rubinstein, A., Seeman, L., Singer, Y.: Locally adaptive optimization: adaptive seeding for monotone submodular functions. In: Proceedings of SODA, pp. 414–429 (2016)
3. Borgs, C., Brautbar, M., Chayes, J., Lucier, B.: Maximizing social influence in nearly optimal time. In: Proceedings of SODA, pp. 946–957 (2014)
4. Chen, W., Peng, B.: On adaptivity gaps of influence maximization under the independent cascade model with full adoption feedback. In: Proceedings of ISAAC, pp. 24:1–24:19 (2019)
5. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of ACM KDD, pp. 199–208 (2009)
6. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of ACM KDD, pp. 1029–1038 (2010a)
7. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of IEEE ICDM, pp. 88–97 (2010b)
8. Chen, W., Peng, B., Schoenebeck, G., Tao, B.: Adaptive greedy versus non-adaptive greedy for influence maximization. In: Proceedings of AAAI (2020)
9. Chen, Y., Krause, A.: Near-optimal batch mode active learning and adaptive submodular optimization. In: Proceedings of ICML, pp. 160–168 (2013)
10. Chen, Y., Hassani, S.H., Karbasi, A., Krause, A.: Sequential information maximization: when is greedy near-optimal? In: Proceedings of COLT, pp. 338–363 (2015)
11. Cuong, N.V., Lee, W.S., Ye, N., Chai, K.M.A., Chieu, H.L.: Active learning for probabilistic hypotheses using the maximum gibbs error criterion. In: Proceedings of NeurIPS, pp. 1457–1465 (2013)
12. Feige, U.: A threshold of ln n for approximating set cover. J. ACM 45(4), 634–652 (1998)
13. Galhotra, S., Arora, A., Roy, S.: Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In: Proceedings of ACM SIGMOD, pp. 743–758 (2016)
14. Golovin, D., Krause, A.: Adaptive submodularity: theory and applications in active learning and stochastic optimization. J. Artif. Intell. Res. 42(1), 427–486 (2011)
15. Goyal, A., Lu, W., Lakshmanan, L.V.S.: SIMPATH: an efficient algorithm for influence maximization under the linear threshold model. In: Proceedings of IEEE ICDM, pp. 211–220 (2011)
16. Han, K., Huang, K., Xiao, X., Tang, J., Sun, A., Tang, X.: Efficient algorithms for adaptive influence maximization. Proc. VLDB Endowment 11(9), 1029–1040 (2018)
17. Hollinger, G.A., Englot, B., Hover, F.S., Mitra, U., Sukhatme, G.S.: Active planning for underwater inspection and the benefit of adaptivity. Int. J. Robot. Res. 32(1), 3–18 (2013)
18. Horel, T., Singer, Y.: Scalable methods for adaptively seeding a social network. In: Proceedings of WWW, pp. 441–451 (2015)
19. Huang, K., Wang, S., Bevilaqua, G., Xiao, X., Lakshmanan, L.V.S.: Revisiting the stop-and-stare algorithms for influence maximization. Proc. VLDB Endowment 10(9), 913–924 (2017)
20. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of ACM KDD, pp. 137–146 (2003)
21. Leskovec, J., Krevl, A.: SNAP datasets: stanford large network dataset collection (2014) http://snap.stanford.edu/data
22. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J.M., Glance, N.S.: Cost-effective outbreak detection in networks. In: Proceedings of ACM KDD, pp. 420–429 (2007)
23. Li, R., Qin, L., Yu, J.X., Mao, R.: Finding influential communities in massive networks. VLDB J. 26(6), 751–776 (2017)
24. Lin, C., Lu, J., Wei, Z., Wang, J., Xiao, X.: Optimal algorithms for selecting top-k combinations of attributes: theory and applications. VLDB J. 27(1), 27–52 (2018)
25. Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, Cambridge (2005)
26. Nguyen, H.T., Thai, M.T., Dinh, T.N.: Stop-and-stare: optimal sampling algorithms for viral marketing in billion-scale networks. In: Proceedings of ACM SIGMOD, pp. 695–710 (2016)
27. Ohsaka, N., Akiba, T., Yoshida, Y., Kawarabayashi, K.: Fast and accurate influence maximization on large networks with pruned monte-carlo simulations. In: Proceedings of AAAI, pp. 138–144 (2014)
28. Ohsaka, N., Sonobe, T., Fujita, S., Kawarabayashi, K.: Coarsening massive influence networks for scalable diffusion analysis. In: Proceedings of ACM SIGMOD, pp. 635–650 (2017)
29. Peng, B., Chen, W.: Adaptive influence maximization with myopic feedback. In: Proceedings of NeurIPS, pp. 3575–3584 (2019)
30. Seeman, L., Singer, Y.: Adaptive seeding in social networks. In: Proceedings of IEEE FOCS, pp. 459–468 (2013)
31. Sun, L., Huang, W., Yu, P.S., Chen, W.: Multi-round influence maximization. In: Proceedings of ACM KDD, pp. 2249–2258 (2018)
32. Tang, J., Tang, X., Yuan, J.: Influence maximization meets efficiency and effectiveness: a hop-based approach. In: Proceedings of IEEE/ACM ASONAM, pp. 64–71 (2017)
33. Tang, J., Tang, X., Xiao, X., Yuan, J.: Online processing algorithms for influence maximization. In: Proceedings of ACM SIGMOD, pp. 991–1005 (2018a)
34. Tang, J., Tang, X., Yuan, J.: An efficient and effective hop-based approach for influence maximization in social networks. Soc. Netw. Anal. Min. 8, 10 (2018b)
35. Tang, J., Huang, K., Xiao, X., Lakshmanan, L.V., Tang, X., Sun, A., Lim, A.: Efficient approximation algorithms for adaptive seed minimization. In: Proceedings of ACM SIGMOD, pp. 1096–1113 (2019)
36. Tang, Y., Xiao, X., Shi, Y.: Influence maximization: Near-optimal time complexity meets practical efficiency. In: Proceedings of ACM SIGMOD, pp. 75–86 (2014)
37. Tang, Y., Shi, Y., Xiao, X.: Influence maximization in near-linear time: A martingale approach. In: Proceedings of ACM SIGMOD, pp. 1539–1554 (2015)
38. Vaswani, S., Lakshmanan, L.V.S.: Adaptive influence maximization in social networks: why commit when you can adapt? arXiv preprint, arXiv:1604.08171 (2016)
39. Yadav, A., Chan, H., Xin Jiang, A., Xu, H., Rice, E., Tambe, M.: Using social networks to aid homeless shelters: dynamic influence maximization under uncertainty. In: Proceedings of AAMAS, pp. 740–748 (2016)
40. Yuan, J., Tang, S.: No time to observe: adaptive influence maximization with partial feedback. In: Proceedings of IJCAI, pp. 3908–3914 (2017)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.