BUDGET-INDEPENDENT INFLUENCE MAXIMIZATION IN NEARLY LINEAR TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Influence maximization asks for k seed vertices that maximize the expected spread of a diffusion process in a network. Standard near-optimal-time algorithms based on reverse-reachable sampling achieve a $(1 - 1/e - \varepsilon)$ approximation, but their expected running-time bounds grow linearly with the seed budget k. We remove this multiplicative dependence: for the independent cascade model, our algorithm succeeds with probability at least $1 - \delta$ in $O((m + n)\varepsilon^{-3} \log(2n/\delta))$ expected time. The result extends to triggering models with explicitly charged local sampling costs. We reserve $O(\varepsilon k)$ seed positions for cost-weighted random vertices, allowing reverse-reachable searches to stop as soon as they encounter a reserved seed. An independent sample-count estimation phase uses a statistic that also controls the expected search cost. Matching these quantities eliminates the multiplicative dependence on k while preserving the approximation guarantee.

1 INTRODUCTION

Influence maximization asks how a limited number of initial activations can reach a large population through network diffusion. It captures the possibility of reaching people beyond those directly recruited, a motivation for network-aware viral marketing (Domingos & Richardson, 2001; Richardson & Domingos, 2002) and peer-led health-information outreach (Wilder et al., 2018). Because different seeds may reach overlapping audiences, the value of one seed depends on the others selected. The task is therefore to allocate a limited seeding budget jointly, rather than simply rank vertices by individual prominence (Kempe et al., 2003; 2015).

Formally, the input consists of a directed graph $G = (V, E)$ with n vertices and m edges, a stochastic diffusion model, and a seed budget k. For a seed set S, let $\sigma(S)$ denote the expected number of activated vertices; the goal is to maximize $\sigma(S)$ subject to $|S| \leq k$. Under the independent cascade (IC) and triggering models, the objective is monotone and submodular, although the optimization problem is NP-hard (Kempe et al., 2003; 2015). The classical greedy analysis yields a $1 - 1/e$ approximation when exact marginal values are available (Nemhauser et al., 1978).

The challenge is to obtain a comparable guarantee efficiently. Repeated marginal-influence evaluations can require many reachability computations over random graph realizations. Reverse influence sampling addresses repeated evaluation by reducing influence maximization to maximum coverage on sampled reverse-reachable (RR) sets (Borgs et al., 2014; Tang et al., 2014). Its standard running-time bounds, however, retain a multiplicative dependence on the seed budget. For instance, the expected time of TIM (Tang et al., 2014) is

$$O\left((m + n)\varepsilon^{-2} \left[k \log n + \log \frac{1}{\delta}\right]\right) \tag{1}$$

for a $(1 - 1/e - \varepsilon)$ approximation with failure probability δ. The revised full analysis of Borgs et al. (2016) also contains a linear dependence on k. These bounds are nearly linear for fixed k, but can become polynomially larger when the budget grows with the network. For example, with fixed accuracy, inverse-polynomial failure probability, and $k = \lfloor \sqrt{n} \rfloor$, the bound in (1) becomes $\widetilde{O}((m + n)\sqrt{n})$ rather than $\widetilde{O}(m + n)$. Eliminating the budget factor would allow larger seed sets without this additional polynomial cost.

We establish a nearly linear expected-time guarantee for every input budget by reducing the cost of generating RR sets. We reserve $O(\varepsilon k)$ seed positions for vertices drawn independently in proportion

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

to their reverse-search costs and require the final solution to contain them. A reverse search can then stop as soon as it hits a reserved seed, because its sample is already covered by the eventual output. Whereas the Hit-and-Stop (HIST) algorithm constructs its stopping set through empirical influence optimization and validation (Guo et al., 2020; 2022), our method uses independent cost-weighted sampling to bound search cost directly. The approximation guarantee relies only on the reserved set being small, not on its influence.

To determine the number of RR samples, we use an independent *sample-count estimation phase* based on a bounded measure of reverse-search cost. Its mean appears inversely in the sample-count bound and linearly in the expected cost bound for each stopped search, so it cancels when the two bounds are combined. Together with the saving from the reserved seeds, this yields the following guarantee.

Theorem 1.1 (Influence maximization in nearly linear time)**.** *In the adjacency-list model, for every IC network,* $1 \le k \le n$*,* $0 < \varepsilon \le 1/4$*, and* $0 < \delta < 1/2$*, there is a randomized algorithm returning* $S \subseteq V$ *with* $|S| \le k$ *such that*

$$\mathbb{P}[\sigma(S) \ge (1 - 1/e - \varepsilon)\,\mathrm{OPT}_k] \ge 1 - \delta, \qquad \mathrm{OPT}_k = \max_{|T| \le k} \sigma(T). \tag{2}$$

Its expected running time is

$$O\left(\frac{m+n}{\varepsilon^3}\left[\ln\frac{en}{k} + \frac{1}{k}\ln\frac{4}{\delta}\right]\right) \subseteq O\left((m+n)\varepsilon^{-3}\log\frac{2n}{\delta}\right). \tag{3}$$

Compared with (1), the theorem removes the multiplicative budget factor while changing the precision dependence from ε^{-2} to ε^{-3}. We develop the analysis for IC and extend it to triggering models with explicit local sampling-cost assumptions in Appendix D. A time-capped version gives a deterministic running-time limit with one additional logarithmic repetition factor.

1.1 Related Work

Origins and approximation framework. Domingos and Richardson introduced network-aware methods for viral marketing (Domingos & Richardson, 2001; Richardson & Domingos, 2002). Kempe, Kleinberg, and Tardos established the combinatorial optimization framework for IC, linear threshold, and more general triggering diffusion, including hardness and submodularity-based approximation results (Kempe et al., 2003; 2015). The underlying cardinality-constrained greedy guarantee is due to Nemhauser et al. (1978).

Simulation-based accelerations and heuristics. CELF, introduced by Leskovec et al. (2007), uses lazy marginal-gain evaluation to accelerate greedy selection; CELF++ of Goyal et al. (2011) further reduces repeated work. These are accelerations of greedy evaluation, rather than substitutes for its objective. A separate line develops fast heuristic influence surrogates. Representative examples are the degree-discount heuristics of Chen, Wang, and Yang (Chen et al., 2009), the local-arborescence methods of Chen et al. (2010), and IRIE of Jung et al. (2012), which combines influence ranking with influence estimation. These methods emphasize practical scalability; they do not by themselves provide the uniform approximation-and-time guarantee proved here.

Reverse sampling and provable scalability. Borgs et al. (2014; 2016) developed reverse influence sampling, and TIM/TIM+ (Tang et al., 2014) separates sample-size estimation from coverage optimization. Subsequent work includes martingale-based estimation in IMM (Tang et al., 2015), online quality assessment in OPIM and OPIM-C (Tang et al., 2018), and improved RR-set generation and tighter complexity bounds by Guo et al. (2020; 2022). The latter works improve the graph-dependent sampling cost, including bounds that retain a factor of k under restrictions on incoming probability sums. Lakshmanan (2025) also studies budget-independent influence maximization, but the guarantee in that paper's Theorem 1 assumes $k\varepsilon < 1$. Our guarantee applies to every $1 \le k \le n$ without this restriction.

Comparison with previous approaches. The most direct precursor to our stopping rule is HIST, introduced by Guo et al. (2020, Section 4) and developed further in Guo et al. (2022, Section 5). HIST first selects a *sentinel set*, meaning a subset of the eventual seeds used to stop subsequent

reverse searches. It then generates RR samples that stop upon hitting this set and selects the remaining seeds. Its sentinel-selection phase uses empirical greedy optimization and sample-based validation to establish an influence guarantee for the selected prefix.

Our method replaces this influence-optimized sentinel construction with $O(\varepsilon k)$ independent cost-weighted draws. The resulting set need not capture any prescribed fraction of the optimal influence: its size controls the approximation loss, while its distribution controls the cost of reverse searches. A prefix-integral argument gives a cost bound inversely proportional to the number of draws plus one. Combining this bound with independent sample-count estimation yields (3).

TIM (Tang et al., 2014, Section 3.2) uses indegree-weighted random seed sets to define an auxiliary influence quantity and controls the inverse moment of the resulting lower-bound estimate. Our cost-weighted draws instead serve as actual seeds that reduce subsequent search work. For sample-count estimation, we apply the accumulated-sum method for bounded observations (Dagum et al., 2000) to a capped search-cost statistic, rather than explicitly estimating an influence lower bound. This statistic can be evaluated exactly without always completing the RR search. Its mean enters the sample-count bound inversely and the per-sample cost bound linearly; canceling these factors links parameter estimation directly to the computational saving from the prescribed seeds.

1.2 Technique Overview

We explain the construction in the order used in the technical sections: first the seed-selection algorithm for a supplied number of samples, and then the phase that determines that number.

From reverse samples to prescribed seeds. A reverse-reachable (RR) set consists of the vertices that can reach a uniformly selected target in one random live-edge graph. A seed set intersects such a sample with probability equal to its influence divided by the number of vertices. Maximizing the fraction of intersected samples is therefore an empirical maximum-coverage problem. Using inverted incidence lists and integer buckets, all greedy selections take time linear in the explicit coverage instance. The difficulty is generating that instance cheaply enough.

As in sentinel-based sampling (Guo et al., 2020; 2022), we require the output to contain a set B before generating the optimization samples, and stop a reverse search as soon as it hits B. Unlike selecting B by empirical influence maximization, we draw at most $\varepsilon k/3$ vertices independently, with probabilities proportional to one plus their indegrees. These weights reflect the cost of expanding a vertex. The membership test precedes the incoming-edge scan, so a costly prescribed vertex stops the search without incurring its own expansion cost. Searches avoiding B are completed, and greedy fills the remaining positions using those samples. Prescribing any set this small loses only $O(\varepsilon)$ in the approximation ratio; no influence-quality certificate for B is needed.

Why the budget factor disappears. Our sufficient sample bound still contains a factor of k, because we require accuracy simultaneously over all feasible seed sets. We remove this factor from the *total work*, not from that accuracy requirement. Under cost-weighted sampling of B, a more expensive search prefix is more likely to contain a prescribed seed. Integrating the probability of avoiding each prefix yields a search-cost bound inversely proportional to the number of prescribed draws plus one. With our rounded ε-fraction of the budget, this contributes $O(1/(\varepsilon k))$. Multiplication with the statistical factor k leaves an additional $1/\varepsilon$, rather than a polynomial budget dependence.

The sample-count estimation phase makes this cancellation valid without knowing the optimum. It measures a bounded statistic of reverse-search cost whose mean controls both the number of final samples and their expected generation cost. A smaller mean allows cheaper searches but calls for more samples; a larger mean has the opposite effect. The unknown mean cancels in their product. Crucially, the preliminary observations are independent of both B and the final sample stream, so conditioning on the chosen sample count preserves the search-cost bound. Accuracy is proved uniformly over all feasible sets before applying it to the data-dependent greedy output.

Computing the statistic without completing the search. The statistic stops increasing once the total cost of reached vertices reaches a threshold equal to a $1/k$ fraction of the graph's total search cost. Its value is then known exactly, so the search can stop before expanding the vertex that crosses the threshold. Repeating these computations until their values sum to a confidence threshold determines

the final sample count. Each computation is charged in proportion to the value it contributes, giving a worst-case bound on the work of this preliminary phase. This second stopping rule is distinct from hitting B: it determines only a scalar needed for sampling, rather than the membership of a complete RR set. Both stopping rules are exact and introduce no additional approximation error.

1.3 Paper Structure

Section 2 defines the input and diffusion models; Appendix A collects the elementary inequalities. Section 3 presents RR sampling, prescribed-seed selection, and approximation for a sufficient sample count. Section 4 determines the count and states the complete running-time guarantee. Proofs are collected in Appendices A–C; Appendix D extends the analysis beyond IC. Section 5 concludes.

2 Preliminaries

The input to influence maximization consists of a finite directed graph $G = (V, E)$, a specified stochastic diffusion model on G, and an integer seed budget $1 \le k \le |V|$. We write $n = |V| \ge 1$, $m = |E|$, $N^-(v) = \{u : (u, v) \in E\}$, and $\deg^-(v) = |N^-(v)|$. Every vertex is eligible as a seed and every seed has unit cost. A diffusion model specifies how activation spreads from an initially active seed set. For a fixed $S \subseteq V$, let $A(S)$ be the random final active set under that model, including the seeds themselves. The influence-maximization problem is to select a set of at most k seeds maximizing expected final spread:

$$\sigma(S) = \mathbb{E}[|A(S)|], \qquad \mathrm{OPT}_k = \max_{S \subseteq V:\, |S| \le k} \sigma(S). \tag{4}$$

Since seeds themselves count as active,

$$k \le \mathrm{OPT}_k \le n. \tag{5}$$

We seek a randomized approximation algorithm that, given accuracy $0 < \varepsilon \le 1/4$ and failure parameter $0 < \delta < 1/2$, returns a feasible set whose influence is at least $(1 - 1/e - \varepsilon)\,\mathrm{OPT}_k$ with probability at least $1 - \delta$. This success probability refers to the algorithm's random choices; the expectation defining σ is over diffusion. We first specify the IC model used in the main analysis, with the graph stored in adjacency lists, and then the triggering model used in the extension.

2.1 Independent cascades and live-edge graphs

An IC network specifies, in addition to G, a probability $p_{uv} \in [0, 1]$ for every edge $(u, v) \in E$. At time zero, exactly the vertices in a chosen set S are active. If u becomes active for the first time at round t, then in round $t + 1$ it has one opportunity to activate each still-inactive out-neighbor v, succeeding with probability p_{uv}. All such edge trials are mutually independent. Each opportunity is used at most once, and active vertices remain active. The process stops when a round activates no new vertex. Since every nonterminal round activates at least one previously inactive vertex, it stabilizes after at most n rounds. Its final active set is denoted by $A(S)$.

A *live-edge graph* for this IC network is the random directed subgraph $L = (V, E_L)$ obtained by drawing independent variables

$$Z_{uv} \sim \mathrm{Bernoulli}(p_{uv}), \qquad (u, v) \in E, \qquad E_L = \{(u, v) \in E : Z_{uv} = 1\}.$$

An edge in E_L is called live. For any fixed directed graph L, define

$$\mathrm{Reach}_L(S) = \{v \in V : \text{there is a directed path in } L \text{ from some } s \in S \text{ to } v\}.$$

Paths of length zero are allowed, so $S \subseteq \mathrm{Reach}_L(S)$.

Under the standard live-edge coupling for IC (Kempe et al., 2015, Section 4.2), $A(S) = \mathrm{Reach}_L(S)$, and hence $\sigma(S) = \mathbb{E}_L[|\,\mathrm{Reach}_L(S)|]$.

2.2 The triggering model

Definition 2.1 (Triggering diffusion (Kempe et al., 2015, Section 4.1))**.** For each vertex v, a triggering model specifies a probability distribution D_v on $2^{N^-(v)}$. Independently across vertices, draw a set

$T_v \sim D_v$ once at the beginning of a diffusion. With $A_0 = S$, the active sets evolve as

$$A_{t+1} = A_t \cup \{v \in V \setminus A_t : T_v \cap A_t \neq \varnothing\}.$$

The corresponding live-edge graph has vertex set V and edge set

$$E_L = \{(u, v) \in E : u \in T_v\}.$$

For fixed triggering sets, induction on t shows that A_t contains exactly the vertices reachable from S by live paths of at most t edges. Thus the final active set is $\mathrm{Reach}_L(S)$, and the objective (4) applies. Dependence among incoming edges of one vertex is allowed; triggering sets at different vertices are independent. IC is the special case in which each $u \in N^-(v)$ is included independently with probability p_{uv}.

3 Reverse-Reachable Sampling with Prescribed Seeds

This section develops the seed-selection phase. We first recall why RR sampling yields an empirical coverage problem, then explain how a small prescribed seed set permits exact early termination. We analyze approximation assuming a sufficiently large sample count N. The only unresolved parameter at the end of the section is how to choose N without knowing OPT_k; Section 4 supplies it.

3.1 The RR-set approach

Sample a live-edge realization L and an independent uniform target $t \in V$. Its reverse-reachable set is

$$R = \mathrm{RR}(L, t) = \{u \in V : u \rightsquigarrow t \text{ in } L\}. \tag{6}$$

A reverse breadth-first search generates R by expanding incoming live edges. The target itself belongs to R, so every RR set is nonempty. A seed set intersects R exactly when it can activate t in this realization. This observation underlies reverse influence sampling (Borgs et al., 2014; Tang et al., 2014). The following identity is due to (Borgs et al., 2016); its proof is included in Appendix B for completeness.

Lemma 3.1 (RR hitting identity (Borgs et al., 2016, Observation 3.2))**.** *For every fixed $S \subseteq V$,*

$$\mathbb{P}[R \cap S \neq \varnothing] = \frac{\sigma(S)}{n}. \tag{7}$$

For N independent RR samples $R_1, \ldots, R_N$, define

$$\widehat{\sigma}_N(T) = \frac{n}{N} \sum_{i=1}^{N} \mathbf{1}\{R_i \cap T \neq \varnothing\}. \tag{8}$$

The ordinary RR algorithm generates the samples, views them as hyperedges, and greedily selects k vertices to cover as many hyperedges as possible. Every selection covers all still-uncovered samples containing the chosen vertex. With sufficiently many samples, approximate maximization of this empirical objective implies approximate maximization of true influence.

Lemma 3.2 (Coverage structure)**.** *Both σ and every fixed empirical objective $\widehat{\sigma}_N$ are normalized, nonnegative, monotone, and submodular.*

The sample count and the cost per sample are separate issues. Even when the explicit coverage instance can be optimized quickly, generating it may require many expensive reverse searches. Our modification changes how much of each sample must be revealed.

3.2 Presampling seeds and stopping covered searches

Instead of starting greedy with no seeds, we first reserve a small number of seed positions for randomly chosen vertices. Let B be the set of distinct vertices obtained. Every eventual output must contain B. Consequently, a reverse search that reaches B has already established that its sample is covered by every possible output. It can return a hit indicator and stop the *entire search*. A search that avoids B must instead return its complete RR set.

The random choice should favor vertices whose expansion is expensive. Under IC, expanding v examines its incoming edges, so its cost is proportional to $1 + \deg^-(v)$. We therefore use the following distribution.

Definition 3.3 (IC search costs and prescribed-seed sampling)**.** For the input IC network, set

$$c(v) = 1 + \deg^-(v), \qquad C = \sum_{v \in V} c(v) = m + n, \qquad \pi(v) = \frac{1 + \deg^-(v)}{m + n}. \tag{9}$$

Choose

$$b = \left\lfloor \frac{\varepsilon k}{3} \right\rfloor \tag{10}$$

and draw b vertices independently with replacement from π. Their set of distinct values is the prescribed seed set B. Thus $|B| \le b$, leaving $k - |B|$ positions for greedy selection.

Algorithm 1 implements the stopped reverse search, denoted AbsorbRR. Its return value HIT records that the complete sample intersects B; we call this an *absorbed sample*. The membership test in line 4 precedes expansion, so a costly prescribed vertex can stop the search without having its own incoming edges scanned.

Algorithm 1: AbsorbRR(B)**: reveal only the relevant part of an RR sample**

Input: The IC network (G, p) and a fixed prescribed set B
Output: HIT, or a complete RR set disjoint from B

1 Sample a uniform root t; initialize a FIFO queue $Q = [t]$ and discovered set $U = \{t\}$
2 **while** $Q \neq \varnothing$ **do**
3 $v \leftarrow \mathrm{pop}(Q)$
4 **if** $v \in B$ **then**
5 **return** HIT ▷ Do not expand v
6 **foreach** $(u, v) \in E$*, in incoming-adjacency order* **do**
7 Draw an independent $Z_{uv} \sim \mathrm{Bernoulli}(p_{uv})$
8 **if** $Z_{uv} = 1$ *and* $u \notin U$ **then**
9 Add u to U and append it to Q
10 **return** U

For a supplied sample count N, SeedSelection (Algorithm 2) calls AbsorbRR (Algorithm 1) exactly N times, keeps the nonhit sets as $\mathcal{R}$, and runs coverage greedy on those sets. A hit consumes one trial and is not replaced. The total trial count remains N, even though only a subset of the trials need stored vertex lists.

Algorithm 2: SeedSelection(k, ε, N)**: seed selection for a supplied sample count**

Input: The IC network and a sampler for π from Definition 3.3; k, ε, and $N \ge 1$
Output: A seed set S of size k

$b \leftarrow \lfloor \varepsilon k/3 \rfloor$
Draw b vertices independently with replacement from π; let B be their set of distinct values
$\mathcal{R} \leftarrow$ an empty multiset
for $i = 1, \ldots, N$ **do**
  $Y \leftarrow$ AbsorbRR(B) using a fresh root and fresh independent edge trials
  if $Y \neq$ HIT **then**
    Append Y to $\mathcal{R}$
$A \leftarrow$ BucketGreedy$(\mathcal{R}, B, k - |B|)$
return $B \cup A$

The implementation of coverage greedy and its total-time analysis are given in Appendix B.2.

3.3 Approximation for a sufficient sample count

We first quantify the loss from prescribing B. This argument is deterministic and holds for every small initial set, not only for a typical random one. Randomness in B is needed later for the running-time analysis.

Lemma 3.4 (Greedy with prescribed seeds). *Let $f : 2^V \to \mathbb{R}_{\geq 0}$ be normalized, monotone, and submodular. Let O be optimal under budget k, and let b be an integer with $0 \leq b \leq k$, and fix $B \subseteq V$ with $|B| \leq b$. Starting from B, repeatedly take an exact maximum-marginal-gain greedy step until the set has size k, obtaining S. Then*

$$f(S) \geq \left(1 - \frac{1}{e}\right)\left(1 - \frac{b}{k}\right) f(O). \tag{11}$$

Applied to the empirical coverage objective, Lemma 3.4 controls the loss from prescribing B. The following theorem also accounts for sampling error; the proof and the empirical-coverage reduction are given in Appendix B.3.

Theorem 3.5 (Sufficient samples for prescribed-seed selection). *Let $0 < \varepsilon \leq 1/4$, $0 < \delta < 1/2$, and*

$$H = \left\lceil k \ln \frac{en}{k} + \ln \frac{4}{\delta} \right\rceil. \tag{12}$$

For a fixed sample count satisfying

$$N \geq \frac{27nH}{\varepsilon^2 \, \mathrm{OPT}_k}, \tag{13}$$

Algorithm 2 returns a $(1 - 1/e - \varepsilon)$ approximation with probability at least $1 - \delta/2$. This holds for every fixed prescribed set of size at most b chosen independently of the final RR samples.

This theorem separates the approximation argument from sample-count estimation. The remaining task is to meet (13) without knowing OPT_k, while controlling the work required to generate those samples.

4 Determining the Sample Count in Nearly Linear Time

The sufficient condition (13) involves the unknown optimum. A direct solution is to estimate that optimum or repeatedly optimize larger sample batches. We instead choose a statistic adapted to the cost of the prescribed-seed searches. Its expectation need not closely approximate OPT_k: the essential requirement is that it support both a sufficient sample count and a matching upper bound on search work.

We first derive this statistic from the search-cost bound. We then show how to observe it cheaply, how a scalar stopping rule determines N, and why the complete algorithm has the claimed approximation and running time. The preliminary searches used to determine the sample count are independent of the prescribed seeds and the RR samples used for seed selection.

4.1 A statistic matched to the cost of reverse searches

For a complete RR set, define

$$w(R) = \sum_{v \in R} c(v), \qquad X(R) = \min\left\{1, \frac{k\, w(R)}{C}\right\}, \qquad \mu = \mathbb{E}_R[X(R)]. \tag{14}$$

The statistic is the normalized search cost truncated at one; we call it the *score* of R. It is small for samples of small total local cost, and equals one when that cost reaches C/k. The algorithm will observe independent copies of $X(R)$ without always revealing all of R.

To see why this statistic is useful, consider a complete reverse search with a fixed vertex order. A costly prefix has a large probability of containing a cost-weighted prescribed seed. Therefore the probability of continuing a long search decreases with accumulated cost. Integrating this survival probability gives the following bound.

Lemma 4.1 (Prefix-integral bound). *Fix a complete search order $v_1, \ldots, v_r$, independent of B, and write $a_i = \sum_{j=1}^{i} c(v_j)$, $a_0 = 0$. Let B consist of the distinct vertices in b independent draws from π. Charge $c(v)$ for each vertex expanded before absorption and let Q_B be the total charge. Then*

$$\mathbb{E}_B[Q_B \mid v_1, \ldots, v_r] \leq \frac{C}{b+1}\left[1 - \left(1 - \frac{a_r}{C}\right)^{b+1}\right]. \tag{15}$$

For a complete RR set, the bracket in Lemma 4.1 is at most $X(R)$ when $b+1 \le k$. Averaging over the RR sample gives the following expected search-cost bound.

Theorem 4.2 (Expected cost of an absorbing sample)**.** *Assume $b+1 \le k$. If the prescribed set B and a fresh RR sample are independent, then*

$$\mathbb{E}_{B,R}[\mathrm{Time}(\mathsf{AbsorbRR}(B))] = O\left(\frac{C\mu}{b+1}\right). \tag{16}$$

Each search has worst-case running time $O(C) = O(m+n)$.

The expectation in (16) is joint over the random prescribed set and the new sample. We do not assert this bound for every fixed B. The final algorithm will keep B independent of the estimation phase, so the random sample count can be analyzed without conditioning on a favorable prescribed set.

4.2 Observing the score by an exact capped search

A complete RR sample may be large, but $X(R) = 1$ as soon as the costs of the vertices already removed from the queue sum to at least C/k. Every such vertex belongs to the complete RR set, so this threshold determines the score exactly. The test includes the current vertex's precomputed cost but precedes the scan of its incoming edges.

Algorithm C.1 in Appendix C.2 implements this rule. It returns 1 when the threshold is reached and otherwise completes the search to compute $kw(R)/C$.

4.3 Accumulating scores to determine the sample count

The estimation phase accumulates independent values $X_1, X_2, \ldots$ returned by $\mathsf{ScoreRR}(k)$. It stops when their sum reaches a prescribed threshold Γ. This inverse-sum rule is a bounded-observation stopping method (Dagum et al., 2000); Lemma 4.3 states its guarantees.

For $\Gamma \ge 1$, let

$$\tau = \min\left\{t : \sum_{i=1}^{t} X_i \ge \Gamma\right\}. \tag{17}$$

The estimation phase returns only τ. It does not return a seed set, an influence estimate, or an estimate of OPT_k.

Lemma 4.3 (Sample-count estimation guarantees)**.** *The stopping time in* (17) *satisfies*

$$\tau \le \left\lceil \frac{\Gamma C}{k} \right\rceil \qquad \textit{deterministically}, \tag{18}$$

$$\mathbb{E}\tau \le \frac{\Gamma+1}{\mu}, \tag{19}$$

$$\mathbb{P}\left[\frac{\Gamma}{2\tau} > \mu\right] \le \exp\left[-\Gamma\left(\ln 2 - \tfrac{1}{2}\right)\right] \le e^{-\Gamma/8}. \tag{20}$$

The estimation phase takes $O(C(\Gamma+1)/k)$ time in the worst case.

Set H as in (12) and define

$$\Gamma = \left\lceil 8 \ln \frac{2}{\delta} \right\rceil, \qquad a_\tau = \min\left\{\frac{n}{k}, \frac{4\tau}{\Gamma}\right\}, \qquad N = \left\lceil \frac{27H}{\varepsilon^2} a_\tau \right\rceil. \tag{21}$$

The next lemma gives the two properties needed from this choice: a sufficient sample count with high probability, and an unconditional moment bound for the running-time analysis.

Lemma 4.4 (Sufficiency and unconditional sample-count bounds)**.** *With probability at least $1 - \delta/2$ over the estimation phase,*

$$a_\tau \ge \frac{n}{\mathrm{OPT}_k}, \qquad N \ge \frac{27nH}{\varepsilon^2 \,\mathrm{OPT}_k}. \tag{22}$$

Unconditionally,

$$\mathbb{E}[a_\tau] \le \min\left\{\frac{n}{k}, \frac{8}{\mu}\right\}, \qquad N \le \left\lceil \frac{27nH}{\varepsilon^2 k} \right\rceil \quad \textit{deterministically}. \tag{23}$$

4.4 THE COMPLETE ALGORITHM: CORRECTNESS AND RUNNING TIME

Algorithm 3 first determines the sample count, then calls SeedSelection(k, ε, N) (Algorithm 2). The calls to ScoreRR, the prescribed seed draws inside SeedSelection, and the final RR realizations use independent randomness. Consequently, after conditioning on the estimation phase, the final sample count is fixed and both the prescribed seeds and the final sample stream retain their intended distributions.

Algorithm 3: BudgetIndependentIM$(G, k, \varepsilon, \delta)$**: the complete algorithm**

Input: An IC network (G, p); $1 \le k \le n$, $0 < \varepsilon \le 1/4$, $0 < \delta < 1/2$
Output: A set of at most k seeds

if $k = n$ **then**
  return V
Build incoming adjacency lists, compute $c(v)$ and $C = m + n$, and initialize a sampler for π and reusable vertex arrays
$H \leftarrow \lceil k \ln(en/k) + \ln(4/\delta) \rceil$; $\Gamma \leftarrow \lceil 8 \ln(2/\delta) \rceil$
$Z \leftarrow 0$; $\tau \leftarrow 0$
while $Z < \Gamma$ **do**
  $Z \leftarrow Z +$ ScoreRR(k) with fresh independent randomness
  $\tau \leftarrow \tau + 1$
$a_\tau \leftarrow \min\{n/k, 4\tau/\Gamma\}$; $N \leftarrow \lceil (27H/\varepsilon^2) a_\tau \rceil$
return SeedSelection(k, ε, N) *using randomness independent of the entire estimation phase*

Correctness.

Theorem 4.5 (Approximation of the complete algorithm)**.** *For the IC model, Algorithm 3 returns a* $(1 - 1/e - \varepsilon)$ *approximation with probability at least* $1 - \delta$*.*

Expected running time and cancellation of the budget. Two sources of randomness must be kept separate. The cost of a final search is averaged jointly over B and that search's edge trials, whereas the number of such searches is determined by the independent sample-count estimation phase. This separation permits a factorization of expected work; it does not require the costs of searches sharing B to be mutually independent.

Theorem 4.6 (Budget-independent time for IC)**.** *Algorithm 3 has expected running time*

$$O\left(\frac{m+n}{\varepsilon^3}\left[\ln\frac{en}{k} + \frac{1}{k}\ln\frac{4}{\delta}\right]\right). \tag{24}$$

The cancellation can now be read directly from the bounds. The number of samples has expectation at most $O(H/(\varepsilon^2\mu))$, while a search costs $O(C\mu/(b+1))$ in joint expectation. Multiplying removes the unknown mean μ. The remaining linear budget term in H is divided by $b + 1 > \varepsilon k/3$, leaving ε^{-3} and logarithmic terms rather than a polynomial dependence on k.

5 CONCLUSION

We developed an influence-maximization algorithm whose nearly linear expected running time has no multiplicative polynomial dependence on the seed budget. The seed-selection phase reserves $O(\varepsilon k)$ positions for cost-weighted random vertices, uses them to terminate already covered reverse searches exactly, and completes the solution by maximum-coverage greedy. Its approximation analysis retains uniform accuracy over all feasible sets. An independent sample-count estimation phase determines a sufficient sample count without guessing the optimum. Exact truncation makes this phase inexpensive, and matching its statistic to the absorbed-search cost eliminates the unknown scalar and the budget factor from the expected-work bound.

The result applies to arbitrary directed IC networks and to triggering models with the declared local sampling access. It concerns a specified cardinality budget with all vertices eligible and equal seed costs. The worst-case dependence on precision remains ε^{-3}; the analysis does not show that this dependence is necessary. Improving it, extending the budget exchange to restricted seed candidates or other constraints, and producing one approximate ordering for all budgets require further ideas.

AI USE STATEMENT

This work was developed through iterative interactions with GPT-6 Astra ultra, which assisted in exploring algorithmic ideas, developing and refining proofs of the approximation and running-time guarantees, examining connections to related work, and drafting and revising the manuscript. The authors have verified the results presented in this paper and take full responsibility for its content and correctness.

REFERENCES

Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *Proceedings of the 25th Annual ACM–SIAM Symposium on Discrete Algorithms*, pp. 946–957. SIAM, 2014. doi: 10.1137/1.9781611973402.70.

Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. arXiv:1212.0884v5, 2016. URL https://arxiv.org/abs/1212.0884v5. Revised full version; cited for the corrected budget-dependent running-time bound.

Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 199–208. ACM, 2009. doi: 10.1145/1557019.1557047.

Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1029–1038. ACM, 2010. doi: 10.1145/1835804.1835934.

Paul Dagum, Richard Karp, Michael Luby, and Sheldon Ross. An optimal algorithm for Monte Carlo estimation. *SIAM Journal on Computing*, 29(5):1484–1496, 2000. doi: 10.1137/S0097539797315306.

Pedro Domingos and Matthew Richardson. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 57–66. ACM, 2001. doi: 10.1145/502512.502525.

Amit Goyal, Wei Lu, and Laks V. S. Lakshmanan. CELF++: Optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th International Conference Companion on World Wide Web*, pp. 47–48. ACM, 2011. doi: 10.1145/1963192.1963217.

Qintian Guo, Sibo Wang, Zhewei Wei, and Ming Chen. Influence maximization revisited: Efficient reverse reachable set generation with bound tightened. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 2167–2181. ACM, 2020. doi: 10.1145/3318464.3389740.

Qintian Guo, Sibo Wang, Zhewei Wei, Wenqing Lin, and Jing Tang. Influence maximization revisited: Efficient sampling with bound tightened. *ACM Transactions on Database Systems*, 47(3):12:1–12:45, 2022. doi: 10.1145/3533817.

Kyomin Jung, Wooram Heo, and Wei Chen. IRIE: Scalable and robust influence maximization in social networks. In *Proceedings of the 12th IEEE International Conference on Data Mining*, pp. 918–923. IEEE Computer Society, 2012. doi: 10.1109/ICDM.2012.79.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146. ACM, 2003. doi: 10.1145/956750.956769.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2015. doi: 10.4086/toc.2015.v011a004.

K. Lakshmanan. Influence maximization independent of seed set size. *Operations Research Letters*, 62:107309, 2025. doi: 10.1016/j.orl.2025.107309.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 420–429. ACM, 2007. doi: 10.1145/1281192.1281239.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14:265–294, 1978. doi: 10.1007/BF01588971.

Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 61–70. ACM, 2002. doi: 10.1145/775047.775057.

Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. Online processing algorithms for influence maximization. In *Proceedings of the 2018 International Conference on Management of Data*, pp. 991–1005. ACM, 2018. doi: 10.1145/3183713.3183749.

Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 75–86. ACM, 2014. doi: 10.1145/2588555.2593670.

Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1539–1554. ACM, 2015. doi: 10.1145/2723372.2723734.

Bryan Wilder, Laura Onasch-Vera, Juliana Hudson, Jose Luna, Nicole Wilson, Robin Petering, Darlene Woo, Milind Tambe, and Eric Rice. End-to-end influence maximization in the field. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1414–1422. International Foundation for Autonomous Agents and Multiagent Systems, 2018. URL `https://www.ifaamas.org/Proceedings/aamas2018/pdfs/p1414.pdf`.