Time-Critical Influence Minimization via Node Blocking

JINGHAO WANG, University of Technology Sydney, Australia
YANPING WU, University of Technology Sydney, Australia
XIAOYANG WANG, The University of New South Wales, Australia
YING ZHANG, University of Technology Sydney, Australia
WENJIE ZHANG, The University of New South Wales, Australia
LU QIN, University of Technology Sydney, Australia

Influence minimization (IMIN) aims to identify a set of nodes to be blocked, such that the expected number of nodes activated by the given seed set is minimized. It has many important applications, such as misinformation suppression, and has been extensively studied in the literature. Existing works for IMIN, however, neglect key temporal information in real-world scenarios. In this paper, we generalize IMIN and study the time-critical influence minimization (TCIM) problem, which aims to minimize the activation duration-aware influence spread of the seed set by a deadline via node blocking. We show that TCIM is NP-hard and APX-hard, and the objective function is non-submodular. To address the problem, we propose CBFM, an efficient and effective algorithm that provides $\tau(1 - 1/e - \epsilon)$-approximation with at least $1 - 3\delta$ probability, where τ is a data-driven parameter, ϵ and δ are tunable error parameters. Novel concentration results are designed to facilitate the establishment of the approximation guarantee. Moreover, we show that CBFM can be extended to tackle the misinformation mitigation (MM) problem. The existing MM solution offers the approximation guarantee only under specific assumptions. Our extended approach is assumption-free yet still attains the same guarantee, thereby bridging the theoretical gap. Finally, we conduct extensive experiments on 11 datasets to validate the performance of proposed algorithms on TCIM, IMIN (a special case of TCIM), and MM problems. The results show that for TCIM, CBFM achieves up to four orders of magnitude speedup over the baseline; for IMIN, CBFM outperforms the state-of-the-art in terms of efficiency, approximation ratio, and memory usage. Moreover, for MM, our solution can be two orders of magnitude faster than the corresponding state-of-the-art.

CCS Concepts: • **Theory of computation → Stochastic approximation**.

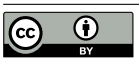
Additional Key Words and Phrases: Influence Minimization; Node Blocking; Non-submodular Optimization

ACM Reference Format:
Jinghao Wang, Yanping Wu, Xiaoyang Wang, Ying Zhang, Wenjie Zhang, and Lu Qin. 2025. Time-Critical Influence Minimization via Node Blocking. *Proc. ACM Manag. Data* 3, 6 (SIGMOD), Article 369 (December 2025), 27 pages. https://doi.org/10.1145/3769834

1 Introduction

The rapid development of the Internet and online social networks (OSNs) enables users to share their perspectives effectively. Leveraging the word-of-mouth effects, information can be extensively propagated within OSNs [2, 4, 14]. However, the vast user base and swift dissemination capabilities of OSNs also accelerate the spread of misinformation, resulting in economic losses and societal

Authors' Contact Information: Jinghao Wang, University of Technology Sydney, Sydney, Australia, jinghao.wang@student.uts.edu.au; Yanping Wu, University of Technology Sydney, Sydney, Australia, yanping.wu@student.uts.edu.au; Xiaoyang Wang, The University of New South Wales, Sydney, Australia, xiaoyang.wang1@unsw.edu.au; Ying Zhang, University of Technology Sydney, Sydney, Australia, ying.zhang@uts.edu.au; Wenjie Zhang, The University of New South Wales, Sydney, Australia, wenjie.zhang@unsw.edu.au; Lu Qin, University of Technology Sydney, Sydney, Australia, lu.qin@uts.edu.au.



ACM 2836-6573/2025/12-ART369
https://doi.org/10.1145/3769834

unrest [2, 34]. As a result, limiting the spread of misinformation has emerged as an important and urgent research task. To this end, one effective strategy is to remove a set of critical nodes (referred to as *blockers*) from the network. Wang et al. [52] formulate this as an optimization problem, termed *influence minimization* (IMIN): given a graph G, a seed set S and a budget k, IMIN is to find a set of k blockers that minimizes the *influence spread* of S (i.e., the expected number of nodes activated by S) under a specified diffusion model.

Existing works on IMIN [42, 51, 52, 59], however, overlook important temporal information. This omission is notable, given that two temporal aspects have been well observed in real-world diffusion [16, 33], and incorporated into other influence-related studies [7, 29, 32, 41, 61]. *First*, diffusion inherently involves time delays [22, 23]. *Second*, the diffusion of information only matters within certain time, i.e., before a deadline [7, 41]. Accounting for these temporal factors in the IMIN setting enables more accurate and realistic analysis. To this end, in the paper, we adopt the *time-aware independent cascade* (TIC) model [7, 29, 30] to capture the diffusion delays between users, and incorporate a *deadline constraint*, aiming to minimize the spread before a given deadline.

Beyond these general aspects, this paper also takes the first step to consider a practical temporal property of misinformation, namely that its impact is stronger on users who are exposed to it earlier. In many real-world scenarios, individuals are permitted to make a single choice at any time within a designated decision period. Consequently, those who encounter misinformation at an earlier stage are more likely to abandon their initial intentions, indicating that misinformation exerts a stronger adverse effect on these individuals. To model this time-sensitive impact, we assign higher weights to nodes that are activated earlier, based on a pre-defined weight function. Building on these foundations, we formulate the *time-critical influence minimization* (TCIM) problem: given a graph G, a seed set S, a budget k, and a deadline T, TCIM is to select k blockers that minimize the total weight over all nodes w.r.t. deadline T under the TIC model. Some applications of TCIM problem are listed below.

- In the context of U.S. presidential election, misinformation (e.g., false claims about a candidate) propagated online can interfere with the election [54]; nevertheless, it is only effective before the election deadline. In addition, in most states (e.g., Florida), once a vote is cast, it is final and cannot be changed [39]. Given that voters may cast their ballot on any one of several voting days, those exposed to misinformation earlier have a higher likelihood of casting a different vote. By blocking the nodes identified by the TCIM solution, voters who are more likely to change their decisions due to misinformation are given higher priority for protection. This approach effectively helps preserve electoral fairness.
- To protect public health, governments often provide free vaccinations, within a limited time [38]. However, misinformation (e.g., claims that vaccines cause harm) can affect vaccination decisions: in a survey of 600 adults, 73.8% of those unexposed to misinformation are vaccinated, compared to only 52.2% of those exposed to six or more misinformation themes [35]. Moreover, individuals may choose to receive a free vaccination on any of the available days. Therefore, earlier exposure to misinformation leads to a higher likelihood of changing one's vaccination decision. To effectively promote vaccine coverage, one method is to block the nodes selected by the TCIM solution.

Challenges and our solution. Although the TCIM problem offers attractive applications, solving it is challenging. In particular, the problem is NP-hard, APX-hard (Theorem 3.3), and the computation of objective function is #P-hard (Theorem 3.4). In view of the hardness, we first propose a baseline that leverages a greedy strategy and Monte-Carlo simulations to optimize the unbiased estimator of objective function. This approach, however, incurs considerable time cost due to the inefficiency of the estimation process. On the other hand, the non-submodularity of TCIM objective (Theorem 3.5) inherently prevents the greedy-based method from attaining a $(1 - 1/e)$-approximation result. Building on the well-established theoretical results in [5], we further conduct a rigorous theoretical

analysis for the baseline. Unfortunately, our analysis reveals that its worst-case approximation guarantee would degrade to 0.

Inspired by the limitations of explicitly optimizing the objective function, in this paper, we propose CBFM, an efficient and effective algorithm with strong approximation guarantee. CBFM employs the sandwich strategy [31] and its basic workflow is simple: it first identifies approximate solutions that maximize the submodular lower and upper bounding functions of the objective function, and then returns the better one among them. However, the development of CBFM is non-trivial due to three main challenges that arise from the newly introduced temporal aspects.

- *Challenge 1:* The effectiveness and approximation ratio of CBFM are highly dependent on the tightness of bounding functions w.r.t. the objective function. On the other hand, submodularity is the key property that enables efficient approximation for the bounding function. Therefore, it is essential to design bounding functions that are not only tight but also preserve submodularity.
- *Challenge 2:* As computing the TCIM objective is #P-hard, our proposed bounding functions are potentially intractable to compute. In fact, we prove their computations are also #P-hard (Lemma 5.1). As a result, unbiased and efficient estimation methods are critical for practical implementation. Existing influence estimation methods [6, 24, 51] focus on the number of activated nodes. However, driven by the weight function, our setting allows non-uniform and real-valued node weights. This fundamental distinction makes existing solutions inapplicable.
- *Challenge 3:* Even though sampling-based estimation methods are designed for the bounding functions, it remains essential to establish the practical sample complexity for CBFM, i.e., mitigate excessive sample generation while ensuring the approximation guarantee of CBFM. To this end, concentration inequalities are required to precisely characterize the deviation between the estimate and the true value. Existing concentration results [47, 51] employed in IMIN and influence maximization studies, are valid when node weights are bounded within $[0, 1]$. This condition, however, does not hold in our setting where node weights can take arbitrary positive values. Moreover, we remark that the Chernoff bound [11] is applicable in theory. Nevertheless, its strict requirement of sample independence will lead to the generation of an impractically large number of samples.

Notably, as detailed in Section 2, the state-of-the-art IMIN algorithm (i.e., SandIMIN) [51] still suffers from several unresolved limitations. This indicates that a direct extension of SandIMIN to TCIM, even if feasible, would still fail to yield a satisfactory solution.

To address Challenge 1, we develop appropriate lower and upper bounding functions for the objective function, with rigorous proofs of their submodularity and monotonicity. In particular, the development of the upper bound is driven by a newly-designed structure named TR tree, which leads to a provably tight approximation to the objective function. The high effectiveness and approximation ratio achieved by CBFM in our experiments further offer empirical validation for the tightness of our bounding functions.

To address Challenge 2, we first propose two new notions of sample sets, namely GRS set and TRS set, in both of which each node is associated with a score. We then devise WS-SSC for their efficient generation. WS-SSC is based on a superior source node selection mechanism WS. Unlike uniformly sampling the source node which may lead to the generation of invalid sample sets, WS enables each generated sample set to be fully utilized for the estimation. A pruning rule is also designed to safely skip the score computation of certain unpromising nodes in the sample set. Finally, with a set of generated random GRS sets (resp. TRS sets), an unbiased estimator for the lower (resp. upper) bounding function is established.

To address Challenge 3, we derive two novel and general concentration inequalities based on a newly constructed martingale sequence, which may be of independent theoretical interest. Leveraging

these results, we develop a stopping condition for CBFM to correctly mitigate the redundant sample generation. Moreover, an update mechanism is devised for efficient blocker selection. We prove that with at least $1 - 2\delta$ probability, CBFM concurrently offers $(1 - 1/e - \epsilon)$-approximate solutions to the lower and upper bounding functions optimizations, where δ and ϵ are the tunable parameters. The better one among them is then returned as the final solution of CBFM, offering a $\tau(1 - 1/e - \epsilon)$-approximation for the TCIM problem with at least $1 - 3\delta$ probability, where τ is a data-driven parameter.

Bridge theoretical gap for the MM problem. We further show that CBFM can be extended to tackle the *misinformation mitigation* (MM) problem. Unlike our blocking-based strategy, MM aims to suppress the spread of misinformation by triggering a competing truth campaign. However, the existing MM solution [40] offers the approximation guarantee only under specific assumptions. Our extended approach resolves this limitation, providing the same approximation guarantee without relying on any assumptions.

Finally, extensive experiments on 11 real-world datasets are conducted to verify the performance of the proposed algorithms. Our results demonstrate that on the TCIM problem, CBFM achieves orders of magnitude speedup over the baseline; on the IMIN problem (a special case of TCIM), CBFM outperforms the state-of-the-art [51] in terms of efficiency, approximation ratio, and memory usage. Furthermore, for the MM problem, our solution can be two orders of magnitude faster than the state-of-the-art [40] due to its advanced stopping condition for sample generation.

Contributions. We summarize our main contribution as follows.

- We formulate the time-critical influence minimization problem. We show that the problem is NP-hard and APX-hard, the objective function is non-submodular, and its computation is #P-hard.
- We design an efficient and effective solution, CBFM, which can provide a $\tau(1 - 1/e - \epsilon)$-approximation for the TCIM problem with at least $1 - 3\delta$ probability.
- We show CBFM can be extended to tackle the misinformation mitigation problem, filling the theoretical gap in existing work.
- We conduct extensive experiments on 11 real-world graphs to evaluate the performance of our algorithms on the TCIM, IMIN, and MM problems. The results demonstrate substantial improvements over the corresponding competitors across all tasks.

Roadmap. The remainder of this paper is organized below. Section 2 reviews related work and shows that adapting existing solutions to solve TCIM is either non-trivial or insufficient to yield satisfactory results. Section 3 introduces the problem definition and proposes a greedy-based baseline. Section 4 develops the bounding functions for TCIM objective and shows the basic workflow of CBFM. Section 5 proposes the method for bounding function estimation. In Section 6, we present the details of CBFM and establish its approximation guarantee. In Section 7, an extended version of CBFM is designed to tackle the MM problem. We conduct extensive experiments in Section 8. Section 9 concludes the paper.

Due to space limitations, all proofs are omitted and provided in Appendix A.4 of the full version [1].

2 Related Work

Graphs have been widely used to represent the relationships of entities in many areas [27, 28, 43–45, 55–58, 62, 63]. The *influence maximization* (IM) problem is a fundamental problem in graph analysis. Kempe et al. [24] first formulate the IM problem, introducing two fundamental diffusion models (i.e., IC and LT models). Since then, the IM problem have been extensively studied in the literature [6, 8–10, 19–21, 37, 46–48, 50, 53]. The state-of-the-art IM solutions [20, 46] with $(1-1/e-\epsilon)$-approximations are built upon the RIS technique [6], a powerful approach for influence estimation. To support the demands of practical scenarios, the time-aware IM problems have also

been extensively studied [7, 12, 18, 25, 29], with most studies considering both edge-level time-delayed propagation and deadline constraint. However, existing time-aware IM solutions are not applicable to address the challenges of TCIM, for two key reasons. First, their objective functions are all submodular while ours is non-submodular. Second, they aim to select a seed set to maximize spread, while TCIM aims to identify a blocker set to minimize the spread of a given seed set.

Due to the important applications in misinformation suppression [2, 34], the *influence minimization* (IMIN) problem has attracted great attention. Wang et al. [52] are the first to formulate IMIN under the IC model and propose a greedy method that incorporates Monte-Carlo simulations to estimate the reduction in influence spread. Xie et al. [59] subsequently propose an improved greedy method that achieves up to six orders of magnitude speedup, by integrating a more efficient estimation method. However, due to the non-submodularity of IMIN under the IC model, the greedy-based algorithms cannot provide any approximation guarantee. Recently, Wang et al. [51] bridge this gap by proposing the first approximation algorithm SandIMIN with provable guarantee, based on the sandwich strategy [31]. Empirically, SandIMIN also achieves up to two orders of magnitude speedup over the method in [59]. Nevertheless, SandIMIN still leaves room for improvement in three respects. First, it yields a suboptimal approximation ratio, as its developed bounding function may significantly deviate from the IMIN objective. Second, it incurs high memory overhead due to a large number of duplicate nodes in its defined sample set. Third, it is not computationally efficient, as it may generate invalid samples that do not contribute to the estimation. Hence, a direct extension of SandIMIN to TCIM, even if feasible, would still be insufficient to yield a satisfactory solution. This paper designs CBFM, an advanced algorithm for TCIM (and IMIN) that alleviates the limitations of SandIMIN by *i*) developing a tighter bounding function, *ii*) defining sample sets that are free of duplicate nodes, and *iii*) avoiding the generation of invalid samples.

In addition, [42, 60] address IMIN under the LT model, yet their solutions' reliance on a submodular objective hinders adaptation to our setting. Moreover, several studies [15, 17, 32, 40, 41, 61] incorporate temporal factors into misinformation suppression by modeling diffusion delays; however, these approaches are also not readily adaptable, as they adopt a competing-truth strategy rather than our blocking strategy to suppress misinformation.

3 Preliminaries

In this paper, we consider a directed graph $G = (V, E)$, where V is the set of n nodes and $E \subseteq V \times V$ denotes the set of m edges. Given an edge $\langle u, v\rangle \in E$, we refer to u as an incoming neighbor of v and v as an outgoing neighbor of u. We use $N_{in}(u)$ and $N_{out}(u)$ to denote the sets of incoming and outgoing neighbors of a node u, respectively. Each edge $\langle u, v\rangle$ is associated with a probability $p(u, v) \in [0, 1]$. Table 1 lists the notations frequently used in this paper. Section 3.1 presents the utilized diffusion model and defines a new metric. Section 3.2 formulates the TCIM problem and shows its hardness. Section 3.3 proposes a baseline and analyzes its limitations.

3.1 Diffusion Model and Metric

Diffusion model. We focus on the *time-aware independent cascade* (TIC) model [7, 29, 30]. Given a seed set S, the TIC diffusion process unfolds in discrete timestamps, as described below. At timestamp 0, all nodes in S are initialized as activated and begin to spread information. Each activated node would not turn back to be inactivated. At each timestamp $t > 0$, each newly activated node u has a single chance to spread information to each of its inactive outgoing neighbors (e.g., v) with probability $p(u, v)$. If the attempt succeeds, the diffusion delay $d_{u,v}$ is then sampled from a distribution $\mathcal{D}$, indicating that v will receive the information at $t + d_{u,v}$. Node v becomes activated at the earliest time it receives the information, referred to as its *activation timestamp* t^v; if v cannot be activated by S, $t^v = +\infty$. This process proceeds until no more nodes can be further activated.

Table 1. Frequently used notations

Notation	Description
$G, G[V\backslash B]$	the directed graph, and its subgraph induced by $V\backslash B$
$w, w[V\backslash B]$	the possible world and its subgraph induced by $V\backslash B$
t^v, t^v_w	the activation timestamp of v in a random diffusion, and the activation timestamp of v in w
$U(\cdot)$	the pre-defined weight function
$\sigma^t_G(S)$	the AD influence spread of S at t
$\mathbb{D}^t_S(B)$	the reduction in $\sigma^t_G(S)$ after removing B
$\mathcal{A}^t_w(S)$	the set of nodes reachable from S before t in w
$\mathbb{I}^*_T(S)$	the expected number of non-seed nodes activated by S before T; $\mathbb{I}^*_T(S) = \mathbb{E}_w[\lvert\mathcal{A}^T_w(S)\rvert] - \lvert S\rvert$
$\mathbb{D}_L(\cdot), \mathbb{D}_U(\cdot)$	lower and upper bounding functions for TCIM objective
$\psi(\cdot)$	the function that can be instantiated as $\mathbb{D}_L(\cdot)$ or $\mathbb{D}_U(\cdot)$
B^o, B^o_L, B^o_U	the size-k optimal solution to $\mathbb{D}(\cdot), \mathbb{D}_L(\cdot)$ and $\mathbb{D}_U(\cdot)$
$\underline{R}, \overline{R}, R$	the random GRS set, TRS set, the generalized sample set that can be instantiated as $\underline{R}$ or $\overline{R}$

Activation duration-aware metric. We denote $t - t^v$ as the *activation duration* of v w.r.t. timestamp t. Below, we introduce a metric named *activation duration-aware (AD) influence spread*, where each node is weighted based on its activation duration.

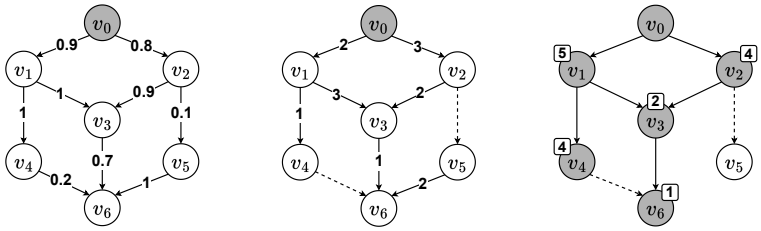
Definition 3.1 (AD influence spread). Given a graph $G = (V, E)$, a seed set S, a timestamp t, and a weight function $U(\cdot) \in [0, +\infty)$, each $v \in V$ is assigned a weight of $\mathbb{E}[U(t - t^v)]$, where the expectation is over the randomness of t^v. The AD influence spread $\sigma^t_G(S)$ of S at t is the total weight over all nodes in V w.r.t. timestamp t, i.e.,

$$\sigma^t_G(S) = \textstyle\sum_{v\in V} \mathbb{E}\left[U(t - t^v)\right].$$

As discussed in Section 1, nodes that receive the misinformation earlier (i.e., with a longer activation duration) should be assigned a larger weight. To formalize this, when $t - t^v > 0$, $U(\cdot)$ is defined as a positive, monotone increasing function w.r.t. $t - t^v$. In addition, to model the deadline constraint, we set $U(t - t^v) = 0$ when $t - t^v \leq 0$. For presentation simplicity, this paper focuses on a unified weight function $U(\cdot)$, shared by each $v \in V$. Nevertheless, it is noteworthy that our analysis also applies to the node-specific setting. In Section 8, two practical instantiations for $U(\cdot)$ (i.e., a unified and a node-specific formulation) are derived to model the real-world scenarios.

Possible world. Alternatively, we can view the stochastic TIC diffusion process by the possible world interpretation. Given a graph $G = (V, E)$, a random possible world w can be viewed as a subgraph of G, where each edge $\langle u, v\rangle \in E$ is removed with $1 - p(u, v)$ probability. The set of edges retained in w is denoted as w_E. Each edge $e \in w_E$ is referred to as a *live* edge and is assigned a diffusion delay d_e randomly drawn from the distribution $\mathcal{D}$. Formally, w is represented as $(V, \mathcal{E})$, where $\mathcal{E} = \{(e, d_e) \mid e \in w_E\}$ is the set of edge-delay pairs. By the possible world interpretation, the AD influence spread can be calculated as $\sigma^t_G(S) = \sum_{v\in V} \mathbb{E}_w\left[U(t - t^v_w)\right]$, where t^v_w denotes the activation timestamp of v in w, and we term $U(t - t^v_w)$ as the weight of v in w w.r.t. timestamp t.

Example 3.2. Figure 1(a) shows a graph G, where v_0 is the seed node, and the number associated with each edge is its propagation probability. Figure 1(b) presents a random possible world w under the TIC model, where the dashed lines represent the removed edges and the number associated with each retained edge denotes its diffusion delay. Figure 1(c) illustrates a diffusion result at timestamp $t = 6$ according to w, where the gray node indicates that it is activated by v_0. The number beside each activated node denotes its weight under w, where we set $U(x) = x + 1$ when $x > 0$.



(a) Graph G (b) Possible world w (c) $t = 6$

Fig. 1. Example of the TIC model

3.2 Problem Definition

In this paper, we study the *time-critical influence minimization* (TCIM) problem. In particular, we consider the *node blocking* strategy, where selecting a node as the blocker implies its removal from G. Given a seed set $S \subseteq V$, a blocker set $B \subseteq V \backslash S$, and a timestamp t, we denote $\mathbb{D}_S^t(B) = \sigma_G^t(S) - \sigma_{G[V\backslash B]}^t(S)$ as the reduction in the AD influence spread of S at t after removing B from G, where $G[V\backslash B]$ is the subgraph of G induced by the node set $V\backslash B$. Equivalently, $\mathbb{D}_S^t(B)$ can be interpreted under the possible world semantics.

$$\mathbb{D}_S^t(B) = \mathbb{E}_w \left[\sum_{v \in \mathcal{A}_w^t(S) \backslash S} \left(U(t - t_w^v) - U(t - t_{w[V\backslash B]}^v) \right) \right],$$

where $\mathcal{A}_w^t(S)$ is the set of nodes activated by S before t under w, and $w[V\backslash B]$ is the subgraph of w induced by $V\backslash B$.

Problem statement. Given a directed graph $G = (V, E)$, a seed set $S \subseteq V$, a budget $k \in \mathbb{Z}^+$, a deadline $T \in \mathbb{Z}^+$, the TCIM problem is to select a size-k blocker set $B^o \subseteq V \backslash S$ that maximizes $\mathbb{D}_S^T(B^o)$ under the TIC model.

For simplicity, we use $\mathbb{D}(\cdot)$ to denote $\mathbb{D}_S^T(\cdot)$ when the context is clear. Note that when $T = +\infty$ and $U(T - t^v) = 1$ holds for any v with $t^v \neq +\infty$, TCIM under the TIC model reduces to the IMIN problem under the *independent cascade* (IC) [24] model.

Problem hardness. We show the hardness results for TCIM below.

Theorem 3.3. *TCIM is NP-hard, and no PTIME method can approximate TCIM within any constant factor $c \in (0, 1)$, unless P=NP.*

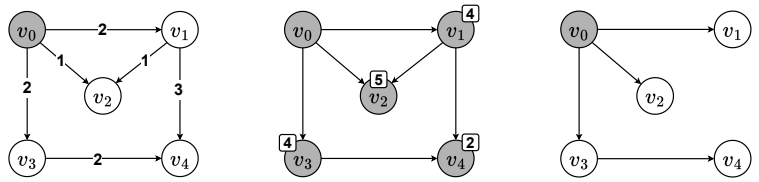
Theorem 3.4. *The computation of $\mathbb{D}(\cdot)$ is #P-hard.*

Moreover, the following theorem shows that $\mathbb{D}(\cdot)$ lacks submodularity, which complicates the design of approximation algorithms.

Theorem 3.5. *$\mathbb{D}(\cdot)$ is monotonic but non-submodular.*

3.3 A Baseline Method

Given the NP-hardness of TCIM and #P-hardness of computing $\mathbb{D}(\cdot)$, we first propose *greedy-based baseline* (GB), which leverages a greedy strategy to optimize the unbiased estimator $\hat{\mathbb{D}}(\cdot)$ of $\mathbb{D}(\cdot)$. Specifically, GB iteratively selects the node x with the largest $\hat{\mathbb{D}}(B \cup \{x\})$ and adds x to the blocker set B, until the budget is exhausted. In addition, a Monte-Carlo method is utilized to compute the unbiased estimator of $\sigma_{G[V\backslash B]}^t(S)$, which in turn enables the computation of $\hat{\mathbb{D}}(B)$. In a nutshell, it performs r simulations. In each simulation, it generates a random possible world $\tilde{w}$ over $G[V\backslash B]$ and computes $\sum_{v \in \mathcal{A}_{\tilde{w}}^T(S)} U(T - t_{\tilde{w}}^v)$ by Dijkstra's algorithm. The average of these values across all simulations serves as an estimator of $\sigma_{G[V\backslash B]}^t(S)$.



(a) Possible world w (b) $t = 5$ (c) TR tree $\mathcal{T}_w$

Fig. 2. Example of proposed bounds

Although GB is simple and effective, it incurs significant computational overhead. In particular, it performs k iterations, each estimating the AD influence spread for $O(n)$ candidate node sets. Additionally, each estimation involves r simulations, with each simulation taking $O(n \log n + m)$ time. As a result, the overall time complexity of GB is $O\left(rkn(n \log n + m)\right)$. On the other hand, due to the non-submodularity of TCIM objective, GB fails to offer $(1 - 1/e)$-approximation result [36]. Bian et al. [5] establish approximation guarantees for the greedy-based method on non-submodular functions, based on the concepts of *curvature* and *submodularity ratio*. Building on their theoretical results, we conduct a rigorous analysis for the approximation guarantee of GB. Unfortunately, our analysis reveals that the submodularity ratio of TCIM objective approaches 0 infinitely in certain scenarios, rendering the approximation ratio of GB as small as 0. Due to the limited space, we defer a detailed explanation to Appendix A.1 of the full version [1].

4 Framework Overview

To develop a practical approximation algorithm, as discussed above, explicitly optimizing the non-submodular TCIM objective may not be a favorable choice. Motivated by this, we adopt the sandwich strategy [31] to optimize the bounding functions of the objective function. Building on this scheme, we propose the *concurrent bounding functions maximization* (CBFM) algorithm for TCIM, offering provable approximation guarantees. In the following, we begin by developing the submodular lower and upper bounding functions for our objective function in Section 4.1 and Section 4.2, respectively. In Section 4.3, we show the basic workflow of CBFM.

4.1 Lower bound of $\mathbb{D}(\cdot)$

We observe that the non-submodularity of $\mathbb{D}(\cdot)$ arises from the joint effect of blockers in B. Specifically, multiple blockers in B may jointly lead to a greater reduction in AD influence spread than the sum of the reductions achieved by considering them individually. For example, in Figure 1(c), blocking either v_1 or v_2 individually does not reduce the weight of v_3 under w; however, blocking them simultaneously reduces v_3's weight to 0. Inspired by this, we develop a submodular lower bounding function $\mathbb{D}_L(\cdot)$ for $\mathbb{D}(\cdot)$, by disregarding the joint effect of blockers in B. In particular, for each activated node v, $\mathbb{D}_L(\cdot)$ only considers its reduced weight caused by an individual blocker in B. If multiple (individual) blockers can lead to a reduction in v's weight, $\mathbb{D}_L(\cdot)$ considers the maximum reduction among them. Formally, $\mathbb{D}_L(\cdot)$ is defined below.

$$\mathbb{D}_L(B) = \mathbb{E}_w\left[\sum_{v \in \mathcal{A}^T_w(S) \setminus S}\left(U(T - t^v_w) - U(T - \max_{b \in B} t^v_{w[V/b]})\right)\right].$$

The following lemmas show that $\mathbb{D}_L(\cdot)$ is a lower bound of $\mathbb{D}(\cdot)$, and possesses both submodularity and monotonicity.

Lemma 4.1. *$\mathbb{D}_L(B) \leq \mathbb{D}(B)$ holds for any $B \subseteq V \setminus S$.*

Lemma 4.2. *$\mathbb{D}_L(\cdot)$ is submodular and monotonic.*

Example 4.3. For ease of exposition, we assume that a certain graph has only one possible world w, as shown in Figure 2(a), where v_0 is the seed and each edge is labeled with its diffusion delay. Figure 2(b) presents the TIC diffusion results at $t = 5$ according to w. The number beside each node denotes its weight under w, where we set $U(x) = x + 1$ when $x > 0$. Let $T = 5$ and $B = \{v_1, v_3\}$. We have $\mathbb{D}(B) = 4 + 4 + 2 = 10$ as v_1 and v_3 jointly protect v_4 from being activated by v_0. Instead, blocking v_3 delays the activation of v_4 from 4 to 5, reducing its weight by 1, while blocking v_1 has no effect on its activation time. Thus we have $\mathbb{D}_L(B) = 4 + 4 + 1 = 9$.

4.2 Upper bounds of $\mathbb{D}(\cdot)$

For each v activated by S in w before T, the weight of v is reduced if and only if its activation timestamp t_w^v increases after blocking B. To establish an upper bound for $\mathbb{D}(\cdot)$, a natural idea is to relax the condition for each node's weight to be reduced. Meanwhile, to preserve submodularity, the joint effect among blockers in B should be avoided. Taking both aspects into account, an upper bounding function for $\mathbb{D}(\cdot)$ can be defined as the expected total weight over all the nodes in $P_w(B)$, where $P_w(B)$ is the subset of $\mathcal{A}_w^T(S)\backslash S$ in which each node has at least one path from S blocked by B in w. Here, a path is considered blocked by B if it includes any node in B. The resulting function is denoted as $\mathbb{D}'_U(\cdot)$ and expressed below.

$$\mathbb{D}'_U(B) = \mathbb{E}_w[\textstyle\sum_{v \in P_w(B)} U(T - t_w^v)].$$

Lemma 4.4 and Lemma 4.5 demonstrate that $\mathbb{D}'_U(\cdot)$ is a submodular and monotonic upper bounding function for $\mathbb{D}(\cdot)$.

Lemma 4.4. *$\mathbb{D}'_U(B) \geq \mathbb{D}(B)$ holds for any $B \subseteq V\backslash S$.*

Lemma 4.5. *$\mathbb{D}'_U(\cdot)$ is submodular and monotonic.*

Tighter upper bound. These desirable properties facilitate the design of approximation algorithms for TCIM. Nevertheless, $\mathbb{D}'_U(\cdot)$ may exhibit substantial deviation from $\mathbb{D}(\cdot)$ in certain scenarios. This primarily stems from the fact that an activated node typically has multiple paths from S in w. For most nodes (e.g., u) in $P_w(B)$, B blocks only one of the paths from S to u in w, and thus the weight of u in w may remain unreduced. Note that, the significant deviation of the bounding function will undermine both the practical effectiveness and the approximation guarantee of CBFM.

To alleviate this issue, we propose a tighter upper bound $\mathbb{D}_U(\cdot)$, which follows the intuition behind the development of $\mathbb{D}'_U(\cdot)$, but incorporates a more fine-grained selection of candidate nodes for accumulating their weights. As the core structure for establishing $\mathbb{D}_U(\cdot)$, *time-aware reachable (TR) tree* is first introduced as follows.

Definition 4.6 (TR Tree). Given a graph $G = (V, E)$, a possible world w, a seed set S and a deadline T, the TR tree of w, represented by $\mathcal{T}_w = (\mathcal{T}_w^V, \mathcal{T}_w^E)$, is defined as follows:

- $\mathcal{T}_w^V = \mathcal{A}_w^T(S)$ is the set of nodes reached by S before T under w.
- $\mathcal{T}_w^E = \{\langle a_v, v\rangle \mid v \in \mathcal{T}_w^V\backslash S\}$. Here, a_v is the incoming neighbor of v in w that activates v at its activation timestamp t_w^v. If multiple incoming neighbors can activate v at t_w^v, the node with the smallest ID is regarded as a_v to break ties, where node IDs are assigned during graph reading and remain fixed thereafter.

In a TR tree $\mathcal{T}_w$, each node v has a unique path from S, which corresponds to a shortest path from S to v in w, i.e., a path whose total edge delay equals t_w^v. With this property, nodes (e.g., u) that are unreachable from B in $\mathcal{T}_w$ fall into one of two cases. First, none of the shortest paths from S to u are blocked by B in w. Second, u has multiple shortest paths from S in w, and B blocks some but not all of them. Due to the tie-breaking rule used in generating the TR tree, the selected path from

S to u in $\mathcal{T}_w$ does not pass through any node in B. In either case, t_w^u is not affected after removing B. As a consequence, we conclude that in any possible world w, the weights of those not reachable from B in $\mathcal{T}_w$ remain unchanged.

Based on this key insight, we define $\mathbb{D}_U(B)$ as the expected total weight of the nodes in $\mathcal{A}_{\mathcal{T}_w}(B)$, the set of nodes reachable from B in $\mathcal{T}_w$. The formal expression of $\mathbb{D}_U(B)$ is given below.

$$\mathbb{D}_U(B) = \mathbb{E}_w[\textstyle\sum_{v \in \mathcal{A}_{\mathcal{T}_w}(B)} U(T - t_w^v)]. \tag{1}$$

The following lemmas show that $\mathbb{D}_U(\cdot)$ is a submodular and monotonic upper bounding function for $\mathbb{D}(\cdot)$ and is tighter than $\mathbb{D}'_U(\cdot)$.

Lemma 4.7. *$\mathbb{D}(B) \le \mathbb{D}_U(B) \le \mathbb{D}'_U(B)$ holds for any $B \subseteq V \backslash S$.*

Lemma 4.8. *$\mathbb{D}_U(\cdot)$ is submodular and monotonic.*

Example 4.9. Continuing with Example 4.3 where $T = 5$ and $B = \{v_1, v_3\}$. Figure 2(c) shows the TR tree $\mathcal{T}_w$ of the possible world w in Figure 2(a). We have $\mathbb{D}(B) = 10, \mathbb{D}'_U(B) = 15$ and $\mathbb{D}_U(B) = 10$.

4.3 Basic Workflow for CBFM

With the proposed bounding functions above, we first formulate two optimization problems, named *lower bounding function maximization* (LBFM) and *upper bounding function maximization* (UBFM). Given a graph G, a seed set S, a budget k and a deadline T, LBFM (resp. UBFM) aims to find a size-k blocker set B_L^o (resp. B_U^o) such that $\mathbb{D}_L(B_L^o)$ (resp. $\mathbb{D}_U(B_U^o)$) is maximized. The basic workflow of CBFM is then presented as follows.

- CBFM first identifies the approximate solutions with performance guarantees for both LBFM and UBFM.
- Among these two solutions, the one that can lead to the larger reduction in $\sigma_G^T(S)$, is returned as the final solution of CBFM.

5 Bounding Functions Estimation

Clearly, the key for CBFM lies in designing efficient and effective approximation algorithms for LBFM and UBFM, both of which are essentially submodular maximization problems. As such, the greedy method offers $(1 - 1/e)$-approximate solutions [36]. Nevertheless, this method is practically infeasible in our setting, as we establish the #P-hardness of computing $\mathbb{D}_L(\cdot)$ and $\mathbb{D}_U(\cdot)$ below.

Lemma 5.1. *The computations of $\mathbb{D}_L(\cdot)$ and $\mathbb{D}_U(\cdot)$ are #P-hard.*

In view of this issue, in this section, we first devise a practical sampling-based method to accurately estimate $\mathbb{D}_L(\cdot)$ and $\mathbb{D}_U(\cdot)$, serving as the foundation for CBFM (detailed in Section 6). In Section 5.1, we introduce two new notions of sample sets. In Section 5.2, we detail the algorithm for the sample sets construction. In Section 5.3, we derive unbiased estimators for $\mathbb{D}_L(\cdot)$ and $\mathbb{D}_U(\cdot)$.

5.1 Sample Sets

We first introduce two sample sets, namely *graph-based reverse score (GRS) set* and *tree-based reverse score (TRS) set*, which form the basis for estimating $\mathbb{D}_L(\cdot)$ and $\mathbb{D}_U(\cdot)$, respectively.

Definition 5.2 (GRS Set & TRS Set). Given a graph G, a seed set S, a source node v, a deadline T and a possible world w, the GRS set $\underline{R}_w(v)$ and TRS set $\overline{R}_w(v)$ of v under w, are both represented as the sets of node-score pairs $(u, d(u))$, where $u \notin S$. In $\underline{R}_w(v)$, each node u is the one that can reach v in w before T, and its score is computed by $d(u) = U(T - t_w^v) - U(t - t_{w[V\backslash u]}^v)$. In $\overline{R}_w(v)$, each node u is the one that can reach v in $\mathcal{T}_w$, with score $d(u) = U(T - t_w^v)$.

Example 5.3. Reconsider Figure 2 with $T = 5$. We have $\underline{R}_w(v_4) = \{(v_1, 0), (v_3, 1), (v_4, 2)\}$ and $\overline{R}_w(v_4) = \{(v_3, 2), (v_4, 2)\}$.

We further define random GRS set $\underline{R}$ (resp. random TRS set $\overline{R}$) as the GRS set (resp. TRS set) of a node randomly selected according to a specific criterion, under a random possible world.

5.2 Random Sample Sets Construction

Generally, constructing a random sample set involves two main tasks: *i*) *source node selection* and *ii*) *sample set materialization*. In this subsection, we first devise the *weighted sampling* (WS) method for source node selection. On this basis, we present the WS-*based sample sets construction* (WS-SSC) algorithm, for simultaneously constructing a random GRS set and a random TRS set.

The WS method. A straightforward and widely used approach for source node selection is *uniform sampling* (US) [6, 51], which selects the source node uniformly at random from a pre-defined sample space (e.g., $V \backslash S$). However, US may not be the optimal choice in our setting, due to the unique nature of TCIM, i.e., only the weights of nodes activated by the seed set S before T can be reduced. More specifically, for a node v randomly sampled by US, S may fail to activate v under a random w, leading to all the scores in $\underline{R} = \underline{R}_w(v)$ and $\overline{R} = \overline{R}_w(v)$ being 0. We refer to such $\underline{R}$ and $\overline{R}$ as invalid. Note that, an invalid $\underline{R}$ (resp. $\overline{R}$) cannot contribute to the estimation of $\mathbb{D}_L(\cdot)$ (resp. $\mathbb{D}_U(\cdot)$) and instead introduces additional overhead.

To resolve this issue, we propose WS, a crucial subroutine that enables WS-SSC to *consistently* generate valid sample sets. At a high level, WS first performs a forward propagation from S until reaching the deadline T, after which it selects the source node uniformly at random from the traversed nodes (excluding S). In particular, the forward propagation follows Dijkstra's algorithm, treating diffusion delay as the distance between nodes and terminating when the shortest distance from S to the current traversed node exceeds T. Nevertheless, it incorporates two key differences:

- It performs in a stochastic manner, i.e., each traversed edge e is first checked for being live based on its propagation probability. If live, it is assigned a diffusion delay d_e sampled from $\mathcal{D}$.
- To facilitate sample set materialization, for each $u \in V$, it maintains additional information: *i*) $AT[u]$, the activation timestamp of u; *ii*) $pred[u]$, the set of incoming neighbors of u that can activate u at $AT[u]$; *iii*) $IsUniqueSP[u]$, with value 1 (resp. 0) indicates that there exists only one (resp. multiple) shortest path from S to u, i.e., the path with length $AT[u]$. It also stores the partial possible world $w' = (V, \mathcal{E}')$, where $\mathcal{E}' = \{(e, d_e) \mid e \in w'_E\}$ is the set of edge-delay pairs and w'_E is the set of live edges during the forward propagation[1].

Example 5.4. Reconsider the diffusion results in Figure 1(c) with v_0 being the seed node and $T = 6$. Unlike US that selects the source node uniformly at random from $\{v_1, v_2, v_3, v_4, v_5, v_6\}$, WS limits its sample space to $\{v_1, v_2, v_3, v_4, v_6\}$.

Details of WS are deferred to Appendix A.2 of the full version [1] due to space constraints.

The WS-SSC method. We then detail our WS-SSC method for sample generation. As shown in Algorithm 1, based on the source node v and the partial possible world w' produced by WS, it proceeds to materialize the sample sets under w', using the TRS-Generation and GRS-Generation procedures. This mechanism guarantees the validity of all generated sample sets by ensuring that v is explicitly activated by S under w', while still preserving the unbiasedness of the estimators, as will be proved in Section 5.3.

Given that the scores of all nodes in $\overline{R}$ are fixed and efficiently computable (see Definition 5.2), to materialize $\overline{R}$, it is sufficient to determine the set of nodes $\overline{R}$ contains (denoted by $V(\overline{R})$), i.e.,

[1]Instead of generating a possible world w by exploring all edges, here we only explore the edges potentially live before T to reveal a partial possible world for efficiency.

Algorithm 1: WS-SSC

Input : The graph $G = (V, E)$, the seed set S, the deadline T.
Output : The random GRS set $\underline{R}$ and random TRS set $\overline{R}$

$(v, AT, pred, IsUniqueSP, w') \leftarrow \mathrm{WS}(G, S, T)$;
$\overline{R} \leftarrow$ TRS-Generation$(v, AT, pred, S)$;
$\underline{R} \leftarrow$ GRS-Generation$(v, AT, pred, IsUniqueSP, w', S)$;
return $(\overline{R}, \underline{R})$;

Algorithm 2: TRS-Generation$(v, AT, pred, S)$

$\overline{R} \leftarrow \emptyset$, $id[\cdot] \leftarrow$ the ID of each node in G, $\overline{v} \leftarrow v$;
while $\overline{v} \notin S$ **do**
    $\overline{R} \leftarrow \overline{R} \cup \{(\overline{v}, U(T - AT[v]))\}, \overline{v} \leftarrow \arg\min_{u \in pred[\overline{v}]} id[u]$;
return $\overline{R}$;

Algorithm 3: GRS-Generation$(v, AT, pred, IsUniqueSP, w', S)$

$\underline{R} \leftarrow \{(v, U(T - AT[v]))\}, \underline{v} \leftarrow v$;
for each $u \in V$ **do** $Compute[u] \leftarrow 0$;
if $IsUniqueSP[v] = 1$ **then**
    while $pred[\underline{v}][0] \notin S$ **do**
        $\underline{v} \leftarrow pred[\underline{v}][0], Compute[\underline{v}] \leftarrow 1$;
else
    Introduce a virtual seed node s, and a virtual edge $\langle s, u \rangle$ with $p(s, u) = 1$ and $d_{s,u} = 0$ for every $u \in S$;
    for each $u \in S$ **do** $pred[u] \leftarrow s$;
    Recursively constructs the subgraph $\mathcal{G}$ from v by following predecessor information in $pred$;
    Apply the Lengauer-Tarjan algorithm on $\mathcal{G}$ to identify the nodes located in all paths from s to v in $\mathcal{G}$ and store them into $\mathcal{B}$;
    for each $u \in \mathcal{B} \setminus (S \cup \{s, v\})$ **do** $Compute[u] \leftarrow 1$;
for each $u \in V \wedge Compute[u] = 1$ **do**
    Invoke the Dijkstra's algorithm on $w'[V \backslash u]$ to obtain $t^v_{w'[V \backslash u]}$;
    $\underline{R} \leftarrow \underline{R} \cup \{(u, U(T - AT[v]) - U(T - t^v_{w'[V \backslash u]}))\}$;
return $\underline{R}$;

those that can reach v in $\mathcal{T}_{w'}$. For this purpose, TRS-Generation (Algorithm 2) iteratively traces back from v based on *pred*; when multiple predecessors exist, the one with the smallest node ID is selected. By the definitions of *pred* and TR tree, the visited nodes (excluding those in S) correspond precisely to $V(\overline{R})$. Hence, all such nodes are added to $\overline{R}$, each assigned a score of $U(T - AT[v])$.

Example 5.5. Reconsider Figure 1(c). Let the deadline $T = 6$ and $U(x) = x+1$ when $x > 0$. Assume that the selected source node is v_6. It can be verified that $U(T - AT[v_6]) = 1$, $pred[v_6] = \{v_3\}$ and $pred[v_3] = \{v_1, v_2\}$. On this basis, TRS-Generation sequentially adds the node–score pairs $(v_6, 1)$, $(v_3, 1)$ and $(v_1, 1)$ into $\overline{R}$.

In contrast, materializing $\underline{R}$ is more challenging, as it requires determining $t^v_{w'[V \backslash u]}$ for each $u \in V(\underline{R})$. A straightforward idea is to execute Dijkstra's algorithm under $w'[V \backslash u]$ for each $u \in V(\underline{R})$. This method, however, does not scale well with increasing $|V(\underline{R})|$.

The GRS-Generation procedure alleviates this issue through an effective *path-based pruning rule*, which enables us to safely skip the score computation for some unpromising nodes. Let $\mathcal{P}_{w'}(S, v) = \{P \mid P \text{ is a path from } u \in S \text{ to } v \text{ with length } t^v_{w'} \text{ in } w'\}$. The pruning technique is based

on the observation that $t^v_{w'}$ increases only if all paths in $\mathcal{P}_{w'}(S, v)$ are blocked under w'. As a result, we compute the scores only for nodes common to all paths in $\mathcal{P}_{w'}(S, v)$; all other nodes are naturally assigned a weight of 0 and can be excluded from $\underline{R}$ to reduce memory overhead.

Algorithm 3 shows the details of GRS-Generation. Initially, each $u \in V$ is assigned a flag $Compute[u]$, indicating whether its score needs to be computed. We then identify the common nodes among all paths in $\mathcal{P}_{w'}(S, v)$, with two cases considered.

- If $IsUniqueSP[v] = 1$, indicating that there exists only one path in $\mathcal{P}_{w'}(S, v)$, then all nodes along this path are exactly the nodes we seek to identify (Lines 3-5).
- Otherwise, we first introduce a virtual seed node s, and a virtual edge $\langle s, u\rangle$ with $p(s, u) = 1$ and $d_{s,u} = 0$ for every $u \in S$. Under such a setting, the nodes (excluding s) shared by all shortest paths from s to v are exactly the common nodes among all paths in $\mathcal{P}_{w'}(S, v)$. We also update $pred[\cdot]$ accordingly by assigning $pred[u] = s$ for every $u \in S$ (Lines 7-9). Next, we construct an auxiliary subgraph $\mathcal{G}$ by tracing backward from v, with edges formed based on the predecessor information in $pred$. Given that $pred$ records the earliest-arriving incoming neighbors for each node, it follows that $\mathcal{G}$ contains only the shortest paths from s to v in w'. Finally, we apply the Lengauer-Tarjan algorithm [26] on $\mathcal{G}$ to determine the set of nodes (excluding s) common to all the paths from s to v in $\mathcal{G}$ (Line 10). By combining the above analysis, it can be verified that the identified nodes are common to all paths in $\mathcal{P}_{w'}(S, v)$.

For these identified nodes (excluding those in S), we repeatedly execute Dijkstra's algorithm to compute their associated scores and store the resulting node-score pairs into $\underline{R}$ (Lines 12-14).

Example 5.6. We reconsider Figure 1(c), following the same settings as in Example 5.5. Since v_6 has two shortest paths from v_0, it follows that $IsUniqueSP[v_6] = 0$. Then, GRS-Generation constructs $\mathcal{G}$, consisting of the node set $\{s, v_0, v_1, v_2, v_3, v_6\}$, and the edge set $\{\langle s, v_0\rangle, \langle v_0, v_1\rangle, \langle v_0, v_2\rangle, \langle v_1, v_3\rangle, \langle v_2, v_3\rangle, \langle v_3, v_6\rangle\}$. Next, it identifies that v_3, v_6 are the nodes that appear on all paths from s to v_6 in $\mathcal{G}$. Given that $U(T - AT[v_6]) - U(T - t^{v_6}_{w[V\setminus v_3]})) = 1 - 0 = 1$, $(v_3, 1)$ is added to $\underline{R}$. Similarly, the pair $(v_6, 1)$ is also added.

Expected time complexity of WS-SSC. Let $p^T(S, v)$ be the probability that S can activate v before T and $|\pi|$ be the number of nodes within a path π. Define $\chi = \mathbb{E}_w[\max_{v\in\mathcal{A}^T_w(S)\setminus S} \min_{\pi\in\mathcal{P}_w(S,v)} |\pi|]$, denoting the expected maximum number of nodes that require score computation. WS-SSC's time complexity is presented below.

Theorem 5.7. *Let v^+ be a node randomly selected from G with probabilities proportional to their out-degrees and α be the inverse function of Ackerman's function, the expected time complexity of WS-SSC is $O(m \log n \cdot (\chi + 1) \cdot \mathbb{E}_{v^+}[p^T(S, v^+)] + m \cdot \alpha(m, n))$.*

Proof Sketch. The first term of complexity comes from WS and up to χ Dijkstra-like traversals (for scores computation); the second term follows from the Lengauer–Tarjan algorithm. □

As many networks (e.g., social networks) exhibit small-world structure, χ is often small. Moreover, we have $\alpha(m, n) \ll \log n$. Thus the running time of WS-SSC behaves like $O(m \log n)$ in practice.
Remark. For practical efficiency, WS-SSC invokes WS only once, sharing the generated partial possible world across both $\underline{R}$ and $\overline{R}$, rather than generating it separately for each. Note that this correlation (between $\underline{R}$ and $\overline{R}$) does not compromise the approximation guarantee of CBFM. More details can be found in Section 6.3.

5.3 Unbiased estimators

To simplify the exposition, we begin by introducing R to unify $\underline{R}$ and $\overline{R}$, and $\psi(\cdot)$ to unify $\mathbb{D}_L(\cdot)$ and $\mathbb{D}_U(\cdot)$. In particular, R and $\psi(\cdot)$ follow a strict one-to-one correspondence, that is, when R is

instantiated as $\underline{R}$ (resp. $\overline{R}$), $\psi(\cdot)$ is uniquely instantiated as $\mathbb{D}_L(\cdot)$ (resp. $\mathbb{D}_U(\cdot)$). We refer to R as the generalized sample set. Given a blocker set B and a generalized sample set R_i, we define the coverage $X_i(B)$ of B on R_i as follows.

$$X_i(B) = \begin{cases} \max_{b \in B \cap V(R_i)} d(b), & \text{if } B \cap V(R_i) \neq \emptyset, \\ 0, & \text{if } B \cap V(R_i) = \emptyset, \end{cases}$$

where $d(b)$ is the score of b in R_i. Define $\mathbb{I}^*_T(S) = \mathbb{E}_w[|\mathcal{A}^T_w(S)|] - |S|$ as the expected number of non-seed nodes activated by S before T. Lemma 5.8 shows a connection between $\psi(B)$ and $X_i(B)$.

Lemma 5.8. *Given a graph $G = (V, E)$, a seed set S, and a generalized sample set R_i, for any $B \subseteq V \backslash S$, we have $\psi(B) = \mathbb{I}^*_T(S) \cdot \mathbb{E}[X_i(B)]$, where the expectation is taken over the randomness of R_i.*

Based on this result, let $\mathcal{R} = \{R_1, R_2, \ldots, R_\theta\}$ be a set of θ generalized sample sets produced by WS-SSC, and $\Lambda_{\mathcal{R}}(B) = \sum_{R_i \in \mathcal{R}} X_i(B)$ be the coverage of B on $\mathcal{R}$, it then follows that $\mathbb{I}^*_T(S) \cdot \Lambda_{\mathcal{R}}(B)/\theta$ offers an unbiased estimator for $\psi(B)$.

6 CBFM

In this section, we present the details for CBFM, which follows the state-of-the-art IM solution (OPIM-C [46]) but is carefully redesigned for TCIM, incorporating three main differences below.

- CBFM adopts a blocker selection method with an elegant update mechanism to enhance efficiency.
- CBFM integrates novel stopping conditions, built on newly derived concentration results, to reduce redundant sampling.
- CBFM concurrently produces the $(1 - 1/e - \epsilon)$-approximate solutions for two optimization problems (i.e., LBFM and UBFM).

In Section 6.1, we detail the blocker selection method. In Section 6.2, we show the general idea of CBFM, followed by its detailed implementation and theoretical analysis in Section 6.3. In Section 6.4, we discuss the superior performance of CBFM on the IMIN problem over the state-of-the-art [51].

6.1 Efficient Blocker Selection

Lemma 5.8 illustrates that the approximate solution for maximizing $\psi(\cdot)$ can be obtained by identifying the blocker set that approximately maximizes its coverage on a given set of generalized sample sets $\mathcal{R}$. With this key insight, we first propose the *blocker selection* (BS) approach to select a blocker set B with the maximum $\Lambda_{\mathcal{R}}(B)$. BS is built on the greedy framework, iteratively selecting the node x with the maximum $\Lambda_{\mathcal{R}}(x|B)$ and adding it to B, until the budget is exhausted, where $\Lambda_{\mathcal{R}}(x|B) = \Lambda_{\mathcal{R}}(B \cup \{x\}) - \Lambda_{\mathcal{R}}(B)$ is the marginal gain of adding x to B w.r.t. the coverage on $\mathcal{R}$. The novelty of BS lies in an elegant update procedure for $\Lambda_{\mathcal{R}}(\cdot|B)$, efficiently handling the new insertion into B at each iteration. Note that, the existing node selection approaches for IM and IMIN [46, 51], which directly remove the samples containing each newly selected node and accordingly update the marginal gains for other nodes to guide the next selection, are not applicable in our case. This is because, for a selected node x within R_i, other nodes in R_i may have a larger score than x and continue to yield positive marginal gains w.r.t. the coverage on R_i, rendering the removal of R_i infeasible. Below, we first present a key notion for our updating mechanism.

Definition 6.1 (Auxiliary set). Given a sample set R_i and node set B, the auxiliary set $R'_i(B)$ of R_i w.r.t. B is defined as $\{(u, X_i(u|B)) \mid u \in V(R_i), X_i(u|B) > 0\}$, where $X_i(u|B) = X_i(B \cup \{u\}) - X_i(B)$ denotes the marginal gain of adding u into B w.r.t. the coverage on R_i. Besides, given a collection $\mathcal{R}$ of sample sets, we denote $\mathcal{R}'(B)$ by a collection of auxiliary sets of each $R_i \in \mathcal{R}$ w.r.t. B.

Given a collection of auxiliary sets $\mathcal{R}'(B \cup \{x\})$, we can efficiently compute $\sum_{R_i \in \mathcal{R}} X_i(u|B \cup \{x\})$, which equals $\Lambda_{\mathcal{R}}(u|B \cup \{x\})$, by retrieving the auxiliary sets containing u within $\mathcal{R}'(B \cup \{x\})$.

Algorithm 4: BS

Input : The set of generalized sample sets $\mathcal{R}$, the budget k and the seed set S.
Output : The blocker set B.

1 $B \leftarrow \emptyset, \mathcal{R}'(B) \leftarrow \mathcal{R}$;
2 Compute $\Lambda_{\mathcal{R}}(u)$ for each $u \in V \backslash S$, by retrieving all the sample sets in $\mathcal{R}$;
3 **for** $1, 2, \ldots, k$ **do**
4 $\quad x \leftarrow \arg\max_{u \in V \backslash S} \Lambda_{\mathcal{R}}(u \mid B)$;
5 $\quad$ **for each** $R'_i(B) \in \mathcal{R}'(B)$ **do**
6 $\quad\quad$ **if** $x \in V(R'_i(B))$ **then**
7 $\quad\quad\quad R'_i(B \cup \{x\}) \leftarrow \emptyset$;
8 $\quad\quad\quad$ **for each** $(u, X_i(u|B)) \in R'_i(B)$ **do**
9 $\quad\quad\quad\quad X_i(u \mid B \cup \{x\}) \leftarrow \max\{X_i(u|B) - X_i(x|B), 0\}$;
10 $\quad\quad\quad\quad$ **if** $X_i(u \mid B \cup \{x\}) > 0$ **then**
11 $\quad\quad\quad\quad\quad R'_i(B \cup \{x\}) \leftarrow R'_i(B \cup \{x\}) \cup (u, X_i(u \mid B \cup \{x\}))$;
12 $\quad\quad\quad\quad \Lambda_{\mathcal{R}}(u|B \cup \{x\}) \leftarrow \Lambda_{\mathcal{R}}(u \mid B) - \min\{X_i(u|B), X_i(x|B)\}$;
13 $\quad\quad\quad \mathcal{R}'(B \cup \{x\}) \leftarrow \mathcal{R}'(B \cup \{x\}) \cup R'_i(B \cup \{x\})$;
14 $\quad\quad$ **else**
15 $\quad\quad\quad \mathcal{R}'(B \cup \{x\}) \leftarrow \mathcal{R}'(B \cup \{x\}) \cup R'_i(B)$;
16 $\quad B \leftarrow B \cup \{x\}$;
17 **return** B;

Inspired by this, our idea is to iteratively maintain $\mathcal{R}'(B \cup \{x\})$ to facilitate the computation of marginal gains.

The pseudocode of BS is outlined in Algorithm 4. We first initialize B as empty, and $\mathcal{R}'(B)$ as $\mathcal{R}$ (Line 1), since $R'_i(\emptyset) = R_i$ holds for each $R_i \in \mathcal{R}$. $\Lambda_{\mathcal{R}}(u)$ is then computed for each $u \in V \backslash S$, by a linear scan for $\mathcal{R}$ (Line 2). Next, BS runs iteratively. In each iteration, we select the blocker x with the largest $\Lambda_{\mathcal{R}}(x|B)$ (Line 4), followed by maintaining $\mathcal{R}'(B \cup \{x\})$ and computing $\Lambda_{\mathcal{R}}(\cdot|B \cup \{x\})$ based on $\mathcal{R}'(B)$. In particular, we only retrieve the auxiliary sets that contain x, while others are directly stored into $\mathcal{R}'(B \cup \{x\})$ (Lines 14-15).

For each $R'_i(B)$ containing x, we first initialize a empty set $R'_i(B \cup \{x\})$ (Lines 6-7). Afterward, for each node u in $R'_i(B)$, based on its associated value $X_i(u|B)$, we compute $X_i(u|B \cup \{x\})$ (Line 9). According to the definition of auxiliary set, if $X_i(u|B \cup \{x\})$ is positive, we insert the pair $(u, X_i(u|B \cup \{x\}))$ into $R'_i(B \cup \{x\})$ (Lines 10-11). $\Lambda_{\mathcal{R}}(u|B \cup \{x\})$ is then updated (Line 12). After processing all the pairs in $R'_i(B)$, $R'_i(B \cup \{x\})$ is stored into $\mathcal{R}'(B \cup \{x\})$ (Line 13) and we proceed to retrieve other auxiliary sets. The following lemma shows the correctness of BS.

Lemma 6.2. *BS correctly computes $\Lambda_{\mathcal{R}}(v|B \cup \{x\})$ for each $v \in V \backslash S$ in each iteration, providing $(1 - 1/e)$-approximation w.r.t. $\Lambda_{\mathcal{R}}(\cdot)$.*

Proof Sketch. We first prove the correctness of Lines 9 and 12 via a case analysis on $X_i(u \mid B) > X_i(x \mid B)$ and $X_i(u \mid B) \leq X_i(x \mid B)$. By the submodularity of $\Lambda_{\mathcal{R}}(\cdot)$, the lemma then holds. □

6.2 General Idea of CBFM

We then show the general idea of CBFM, with pseudocode given in Algorithm 5. To start, we define $\theta_{\max}$ as the maximum number of samples required to ensure the desired approximation, and θ as the initial sample size (Line 1). Next, by employing WS-SSC (Algorithm 1), we generate two distinct collections of GRS sets $\underline{\mathcal{R}}_1$, $\underline{\mathcal{R}}_2$, and two distinct collections of TRS sets $\overline{\mathcal{R}}_1$, $\overline{\mathcal{R}}_2$, each with a size of θ (Line 2). Subsequently, CBFM runs in an iterative manner to mitigate the generation of excessive sample sets. At each iteration, we first invoke BS (Algorithm 4) on $\underline{\mathcal{R}}_1$ (resp. $\overline{\mathcal{R}}_1$) to obtain

Algorithm 5: CBFM

Input : The graph G, the seed set S, the budget k, the deadline T and the error parameters ϵ, δ.
Output : The blocker set B

1 $\theta_{\max} \leftarrow$ Eq. (5), $OPT_L \leftarrow$ Eq. (4), $\theta \leftarrow \theta_{\max} \cdot \frac{\epsilon^2 OPT_L}{n \cdot U(T-1)}$, $i_{\max} \leftarrow \lceil \log_2 \frac{\theta_{\max}}{\theta} \rceil$;
2 Generate two sets $\underline{\mathcal{R}}_1$ and $\underline{\mathcal{R}}_2$ of random GRS sets with $|\underline{\mathcal{R}}_1| = |\underline{\mathcal{R}}_2| = \theta$, and two sets $\overline{\mathcal{R}}_1$ and $\overline{\mathcal{R}}_2$ of random TRS sets with $|\overline{\mathcal{R}}_1| = |\overline{\mathcal{R}}_2| = \theta$;
3 **for** $1, 2, \ldots, i_{\max}$ **do**
4 $B_L \leftarrow \mathrm{BS}(\underline{\mathcal{R}}_1, k, S), B_U \leftarrow \mathrm{BS}(\overline{\mathcal{R}}_1, k, S)$;
5 **if** $i = i_{\max}$ **then break**;
6 Compute ∇_L and ∇_U on $\underline{\mathcal{R}}_2$ and $\overline{\mathcal{R}}_2$, by Lemma 6.6 with $\alpha = \ln \frac{3 i_{\max}}{\delta}$;
7 **if** $\nabla_L \geq 1 - 1/e - \epsilon \wedge \nabla_U \geq 1 - 1/e - \epsilon$ **then break**;
8 Double the sizes of $\underline{\mathcal{R}}_1$ and $\underline{\mathcal{R}}_2$ with new random GRS sets, and double the sizes of $\overline{\mathcal{R}}_1$ and $\overline{\mathcal{R}}_2$ with new random TRS sets;
9 $\hat{\sigma}^T_{G[V\setminus\cdot]}(S) \leftarrow$ the (ϵ, δ)-estimate of $\sigma^T_{G[V\setminus\cdot]}(S)$;
10 $B^* \leftarrow \arg\min_{B \in \{B_L, B_U\}} \hat{\sigma}^T_{G[V\setminus B]}(S)$;
11 **return** B^*;

candidate blocker set B_L (resp. B_U) (Line 4). To verify whether both B_L and B_U achieve the desired approximation guarantees, we compute ∇_L and ∇_U on $\underline{\mathcal{R}}_2$ and $\overline{\mathcal{R}}_2$ respectively (Line 6). Here, ∇_L (resp. ∇_U) is the lower bound of the approximation ratio achieved by B_L (resp. B_U) w.r.t. $\mathbb{D}_L(\cdot)$ (resp. $\mathbb{D}_U(\cdot)$). If ∇_L and ∇_U both exceed $1 - 1/e - \epsilon$, then B_L and B_U are considered satisfactory, and the iteration terminates (Line 7); otherwise, we double the sizes of $\underline{\mathcal{R}}_1, \underline{\mathcal{R}}_2, \overline{\mathcal{R}}_1$ and $\overline{\mathcal{R}}_2$, and continue to the next iteration (Lines 9-10). Finally, by instantiating the stopping rule algorithm in [64], we calculate the (ϵ, δ)-estimates[2] of $\sigma^T_{G[V\setminus B_L]}(S)$ and $\sigma^T_{G[V\setminus B_U]}(S)$, and return the one with the smaller estimated value among $\{B_L, B_U\}$ (Lines 9-11). Due to space limitations, the details for the $\sigma^T_{G[V\setminus\cdot]}(S)$ estimation are deferred to Appendix A.3 of the full version [1].

6.3 Detailed Implementation and analysis

To make CBFM work efficiently and correctly, it is essential to establish the appropriate parameters (i.e., $\theta_{\max}$, ∇_L and ∇_U). To this end, in what follows, we first propose two novel concentration bounds tailored for describing how close between $\mathbb{I}^*_T(S) \cdot \Lambda_{\mathcal{R}}(\cdot)/\theta$ and $\psi(\cdot)$ for a given $\mathcal{R}$. To our knowledge, the existing concentration results (e.g., Chernoff bounds [11], martingale-based bounds [40, 47, 51]) are not applicable here. This is because, Chernoff bounds require all the samples within $\mathcal{R}$ to be independent, which contradicts the dependencies among the samples caused by the stopping condition of CBFM (Line 7 of Alg. 5). Additionally, since the node weights in our setting can fall in any positive range, existing martingale-based bounds will lead to biased estimates.

Our concentration bounds. To allow the dependencies among all the sample sets, our concentration bounds are also derived based on the *martingale* [11], with the definition shown below.

Definition 6.3 (Martingale). A sequence of random variables $M_1, M_2, \ldots, M_i$ is a martingale, if and only if $\mathbb{E}[|M_i|] < +\infty$ and $\mathbb{E}[M_i \mid M_1, M_2, \ldots, M_{i-1}] = M_{i-1}$ for any i.

In particular, our novelty lies in the construction of the martingale. For a generalized sample set R_i and blocker set B, we first define $Y_i(B) = X_i(B)/U(T-1)$. When the context is clear, $Y_i(B)$ will be simplified to Y_i. Besides, for a set of θ generalized sample sets $R_1, R_2, \ldots, R_\theta$, let $p = \frac{\psi(B)}{\mathbb{I}^*_T(S) \cdot U(T-1)}$ and $Z_i = \sum_{j=1}^{i}(Y_j - p)$. Then, by i) proving $Z_1, Z_2, \ldots, Z_\theta$ form a martingale; ii) deriving the upper

[2]We call $\hat{\mu}$ is the (ϵ, δ)-estimate of μ if $\hat{\mu}$ satisfies $\Pr[(1-\epsilon)\mu \leq \hat{\mu} \leq (1+\epsilon)\mu] \geq 1 - \delta$.

bound for Z_1 and $Z_k - Z_{k-1}$ ($\forall k \in [2, \theta]$); *iii*) analyzing the variance of Y_i ($\forall i \in [1, \theta]$); (*iv*) using the concentration results for martingale [11], we derive the following concentration bounds.

Lemma 6.4. *Given a blocker set B and a set of θ generalized sample sets $\mathcal{R} = \{R_1, R_2, \ldots, R_\theta\}$. For any $\lambda > 0$, we have*

$$\Pr\left[\frac{\Lambda_{\mathcal{R}}(B)}{U(T-1)} - \frac{\theta \cdot \psi(B)}{\mathbb{I}^*_T(S)U(T-1)} \geq \lambda\right] \leq \exp\left(-\frac{\lambda^2}{\frac{2\lambda}{3} + \frac{2\theta \cdot \psi(B)}{\mathbb{I}^*_T(S)U(T-1)}}\right), \tag{2}$$

$$\Pr\left[\frac{\Lambda_{\mathcal{R}}(B)}{U(T-1)} - \frac{\theta \cdot \psi(B)}{\mathbb{I}^*_T(S)U(T-1)} \leq -\lambda\right] \leq \exp\left(-\frac{\mathbb{I}^*_T(S)U(T-1)\lambda^2}{2\theta \cdot \psi(B)}\right). \tag{3}$$

Deriving $\theta_{\max}$. Then, on the basis of the proposed concentration bounds, we derive the setting of $\theta_{\max}$ below, which guarantees the approximation ratio of the results in iteration $i_{\max}$.

Lemma 6.5. *Let $\mathcal{R}_1$ be a set of generalized sample sets with $|\mathcal{R}_1| \geq \theta_{\max}$, B be the size-k blocker set returned by applying BS on $\mathcal{R}_1$, and $N^k_T(S)$ be the set of k nodes in $N_{out}(S)$ with the largest probabilities of being activated by S before T. For fixed $\epsilon, \delta \in (0, 1)$, by setting*

$$OPT_L = U(1) \cdot \textstyle\sum_{v \in N^k_T(S)} p^T(S, v), \tag{4}$$

$$\theta_{\max} = \frac{2n \cdot U(T-1)\left((1-1/e)\sqrt{\ln\frac{6}{\delta}} + \sqrt{(1-1/e)\left(\ln\binom{n-|S|}{k} + \ln\frac{6}{\delta}\right)}\right)^2}{\epsilon^2 \cdot OPT_L}, \tag{5}$$

then B is a $(1 - 1/e - \epsilon)$-approximate solution w.r.t. $\psi(\cdot)$, with at least $1 - \delta/3$ probability.

Deriving ∇_L and ∇_U. Let B^o_L and B^o_U be the size-k optimal solutions for the LBFM and UBFM problems, respectively. Building on our concentration bounds as well, we derive ∇_L and ∇_U below.

Lemma 6.6. *For any fixed α, by defining*

$$\nabla\left(\mathcal{R}, B, U(T-1)\right) = \frac{\left(\sqrt{\Lambda_{\mathcal{R}}(B)/U(T-1) + 2\alpha/9} - \sqrt{\alpha/2}\right)^2 - \alpha/18}{\left(\sqrt{\Lambda_{\mathcal{R}}(B)/(1-1/e)/U(T-1) + \alpha/2} + \sqrt{\alpha/2}\right)^2}, \tag{6}$$

$$\nabla_L = \nabla(\underline{\mathcal{R}}_2, B_L, U(T-1)), \nabla_U = \nabla(\overline{\mathcal{R}}_2, B_U, U(T-1)) \tag{7}$$

we have $\Pr\left[\frac{\mathbb{D}_L(B_L)}{\mathbb{D}_L(B^o_L)} \leq \nabla_L \wedge \frac{\mathbb{D}_U(B_U)}{\mathbb{D}_U(B^o_U)} \leq \nabla_U\right] \leq 4\exp(-\alpha)$.

Approximation guarantee of CBFM. With Lemma 6.6, by setting $\alpha = \ln\frac{3i_{\max}}{\delta}$ in Line 7 of CBFM, $\frac{\mathbb{D}_L(B_L)}{\mathbb{D}_L(B^o_L)} \geq \nabla_L$ and $\frac{\mathbb{D}_U(B_U)}{\mathbb{D}_U(B^o_U)} \geq \nabla_U$ hold simultaneously with at least $1 - 4\delta/3i_{\max}$ probability. This ensures that, in each of the first $i_{\max} - 1$ iterations, if both ∇_L and ∇_U exceed $1 - 1/e - \epsilon$, then with at least $1 - 4\delta/3i_{\max}$ probability, B_L and B_U achieve the $(1 - 1/e - \epsilon)$-approximations w.r.t. $\mathbb{D}_L(\cdot)$ and $\mathbb{D}_U(\cdot)$, respectively. In the last iteration $i_{\max}$ where $|\underline{\mathcal{R}}_1| = |\overline{\mathcal{R}}_1| \geq \theta_{\max}$, as proved in Lemma 6.5, BS returns desired solutions B_L and B_U with at least $1 - 2\delta/3$ probability. By the union bound, it follows that with at least $1 - 2\delta$ probability, CBFM concurrently offers $(1 - 1/e - \epsilon)$-approximate solutions for the LBFM and UBFM problems.

On this basis, the approximation guarantee of CBFM for the TCIM problem is established in the following theorem.

Theorem 6.7. *Given $0 \leq \epsilon, \delta, \leq 1$, CBFM returns B^* satisfies:*

$$\Pr[\mathbb{D}(B^*) \geq \tau(1 - 1/e - \epsilon) \cdot \mathbb{D}(B^o)] \geq 1 - 3\delta. \tag{8}$$

where $\tau = \max\left\{\frac{\mathbb{D}_L(B^o_L)}{\mathbb{D}(B^o)}, \frac{\mathbb{D}(B_U)}{\mathbb{D}_U(B_U)}\right\} \cdot \frac{1-\epsilon}{1+\epsilon}$ is a data-driven parameter and B^o is the size-k optimal solution for the TCIM problem.

Expected time complexity of CBFM. Moreover, the expected time complexity of CBFM is presented below in Theorem 6.8.

Theorem 6.8. *Let EPT be the expected time complexity of WS-SSC shown in Theorem 5.7, and* $v^* = \arg\max_{v \in V} \mathbb{I}_T^*(v)$. *When* $\delta \leq 1/2$, *the expected time complexity of CBFM is*

$$O\left(\frac{EPT \cdot U(T) \cdot \max\{\mathbb{I}_T^*(S), \mathbb{I}_T^*(v^*)+1\} \cdot (k^2 \ln n + k \ln(1/\delta) + \log n \cdot \ln(1/\delta))}{\epsilon^2 \cdot \min\{\mathbb{D}_L(B_L^o), \sigma_G^T(S) - \mathbb{D}(B^o)\}}\right).$$

Proof Sketch. The running time comprises three parts: (i) generating the samples (i.e., invocations of WS-SSC); (ii) applying BS and computing ∇_L, ∇_U over all iterations; and (iii) the final (ϵ, γ)-estimation step. Combining these gives the claimed complexity. □

As shown in the derived complexity expression, the term $\Psi = \frac{U(T) \cdot \max\{\mathbb{I}_T^*(S), \mathbb{I}_T^*(v^*)+1\}}{\min\{\mathbb{D}_L(B_L^o), \sigma_G^T(S) - \mathbb{D}(B^o)\}}$ is hard to compute exactly in practice. To better understand the practical significance of our result, we first provide the approximate value of Ψ in general cases.

$$\Psi \approx \frac{\sigma_G^T(S)}{\min\{\mathbb{D}(B^o), \sigma_G^T(S) - \mathbb{D}(B^o)\}} = \frac{1}{\min\{\rho, 1-\rho\}},$$

where $\rho = \mathbb{D}(B^o)/\sigma_G^T(S)$ denotes the reduction ratio caused by B^o. In addition, as analyzed before, the running time of WS-SSC behaves like $O(m \log n)$, i.e., $EPT \approx O(m \log n)$. Therefore, the time complexity of CBFM can be approximated as

$$O\left(\frac{m \log n \cdot (k^2 \ln n + k \ln(1/\delta) + \log n \cdot \ln(1/\delta))}{\epsilon^2 \cdot \min\{\rho, 1-\rho\}}\right).$$

Unless the budget is particularly limited, ρ does not take very small values. On the other hand, achieving a very large ρ is often infeasible under realistic budget constraints. This suggests that the term $\frac{1}{\min\{\rho, 1-\rho\}}$ remains moderate in practice. As such, in most cases, the time complexity of CBFM is nearly linear in m, indicating its superior scalability.

6.4 Comparsion with SandIMIN

As discussed in Section 3.2, IMIN is a special case of TCIM that does not consider temporal factors. Thus, CBFM is also applicable to the IMIN problem. In this context, we remark that CBFM exhibits superior performance over the state-of-the-art algorithm SandIMIN [51], which also employs the sandwich strategy to optimize the bounding functions of IMIN non-submodular objective function. Specifically, CBFM outperforms SandIMIN in three aspects.

- CBFM provides a **stronger approximation guarantee** by leveraging the newly designed TR tree to construct a tighter upper bounding function, which directly determines the approximation ratio of the sandwich-based approach.
- CBFM exhibits **higher efficiency** by ensuring that each generated sample set is fully utilized for estimation, avoiding the invalid sample sets frequently produced by SandIMIN.
- CBFM exhibits **lower memory usage** by defining lightweight, duplicate-free sample sets, effectively eliminating the redundant nodes present in SandIMIN's sample sets.

Section 8 shows empirical evidence that CBFM surpasses SandIMIN in all the above aspects for the IMIN problem.

7 Extension for MM

This section further demonstrates that CBFM can be extended to tackle the *misinformation mitigation* (MM) problem [40]. We first revisit the MM problem and uncover the theoretical gap in the existing solution. We then extend our techniques to bridge this gap.

Revisit the MM problem. In contrast to our blocking-based strategy, the MM problem aims to suppress the spread of misinformation by triggering a competing truth campaign. MM focuses on the TCIC model, which captures the widely observed tendency of misinformation to spread faster than truth in social networks [49]. In addition, a metric is defined to evaluate the effectiveness of the misinformation mitigation. Specifically, given a possible world w, a misinformation seed set S_m and a truth seed set S_t, each node v is associated with a reward $r_w(S_t, v) \in \{0, 1, 2\}$, which is determined by the delay between the arrivals of the two campaigns at v under w. An earlier arrival of the truth campaign relative to the misinformation campaign results in a higher reward. Given a graph $G = (V, E)$, a misinformation seed set S_m and a budget k, the MM problem is to select a size-k truth seed set $S_t^o \in V \backslash S_m$ such that $\mu(S_t^o)$ is maximized under the TCIC model, where $\mu(S_t^o) = \sum_{v \in V} \mathbb{E}_w[r_w(S_t^o, v)]$ denotes the total expected reward induced by S_t^o.

The MM objective $\mu(\cdot)$ is proven to be non-submodular. In light of this, Simpson et al. [40] design a sandwich-based approach SA. It first derives the submodular lower and upper bounding functions, denoted by $\mu_l(\cdot)$ and $\mu_u(\cdot)$, respectively. Then, it defines a general notion termed *reverse delayed reward (RDR) set*, in which each node is assigned a value from $\{0, 1, 2\}$. Let $\mathcal{R}_l$ and $\mathcal{R}_u$ be the collections of random RDR sets generated with respect to $\mu_l(\cdot)$ and $\mu_u(\cdot)$, respectively. For any truth seed set S_t, Simpson et al. [40] prove that $\mathbb{I}^*(S_m) \cdot \frac{\Lambda_{\mathcal{R}_l}(S_t)}{|\mathcal{R}_l|}$ (resp. $\mathbb{I}^*(S_m) \cdot \frac{\Lambda_{\mathcal{R}_u}(S_t)}{|\mathcal{R}_u|}$) is an unbiased estimator for $\mu_l(S_t)$ (resp. $\mu_u(S_t)$), where $\mathbb{I}^*(S_m)$ is the expected number of non-seed nodes activated by S_m and Λ is the coverage function defined in Section 5.3. Building on the estimators, the NAMM method is designed to produce solutions that maximize $\mu_l(\cdot)$ and $\mu_u(\cdot)$. The better solution among them is returned as the final solution of SA.

Although SA exhibits high efficacy, we note that its approximation guarantee is assumption-dependent, as the concentration results employed by NAMM rely on the assumption that the $(\epsilon/2, \delta/9)$-estimate for $\mathbb{I}^*(S_m)$ is less than $n/2$. This may cause SA to degenerate into a heuristic.

Our solution for MM. Motivated by the observation that the reward $r_w(S_t, v)$ can be considered as a node weight, we extend CBFM to tackle the MM problem. From an algorithmic perspective, our approach (termed SA+) differs from CBFM in three aspects. First, SA+ maintains collections of random RDR sets: $\mathcal{R}_l^1$ and $\mathcal{R}_l^2$ with respect to $\mu_l(\cdot)$, and $\mathcal{R}_u^1$ and $\mathcal{R}_u^2$ with respect to $\mu_u(\cdot)$. Second, it invokes BS (i.e., Alg. 4) on $\mathcal{R}_l^1$ and $\mathcal{R}_u^1$ to identify the candidate truth seed sets S_t^l and S_t^u, respectively. Third, it adopts different stopping conditions for sample generation, i.e., it defines $\nabla_L = \nabla(\mathcal{R}_l^2, S_t^l, 2)$,$\nabla_U = \nabla(\mathcal{R}_u^2, S_t^u, 2)$ (where ∇ is defined in Eq. (6)), and

$$\theta_{\max} = \frac{2n\left((1-1/e)\sqrt{\ln\frac{6}{\delta}} + \sqrt{(1-1/e)\left(\ln\binom{n-|S_m|}{k} + \ln\frac{6}{\delta}\right)}\right)^2}{\epsilon^2 \cdot LB},$$

where LB is defined in [40], denoting a lower bound for the optimal value of the MM objective with budget k. To establish the approximation guarantee of SA+, we derive *assumption-free* concentration bounds that characterize how closely $\mu_g(\cdot)$ approximates $\mathbb{I}^*(S_m) \cdot \Lambda_{\mathcal{R}_g}(\cdot)/|\mathcal{R}_g|$, given a collection of sample sets $\mathcal{R}_g$. Here, $\mu_g(\cdot)$ and $\mathcal{R}_g$ serve as unified notations: when $\mu_g(\cdot)$ is instantiated as $\mu_l(\cdot)$ (resp. $\mu_u(\cdot)$), $\mathcal{R}_g$ corresponds uniquely to $\mathcal{R}_l$ (resp. $\mathcal{R}_u$).

Lemma 7.1. *Given a misinformation seed set S_m, a truth seed set S_t, and a set of θ sample sets $\mathcal{R}_g$. For any $\lambda > 0$, we have*

$$\Pr\left[\frac{\Lambda_{\mathcal{R}_g}(S_t)}{2} - \frac{\theta \cdot \mu_g(S_t)}{2\mathbb{I}^*(S_m)} \geq \lambda\right] \leq \exp\left(-\frac{\lambda^2}{\frac{2\lambda}{3} + \frac{\theta \cdot \mu_g(S_t)}{\mathbb{I}^*(S_m)}}\right), \tag{9}$$

$$\Pr\left[\frac{\Lambda_{\mathcal{R}_g}(S_t)}{2} - \frac{\theta \cdot \mu_g(S_t)}{2\mathbb{I}^*(S_m)} \leq -\lambda\right] \leq \exp\left(-\frac{\mathbb{I}^*(S_m)\lambda^2}{\theta \cdot \mu_g(S_t)}\right). \tag{10}$$

Table 2. Statistics of datasets

Dataset	Type	$\|V\|$	$\|E\|$	Avg. deg
EmailCore (EC)	directed	1,005	25,571	50.9
Facebook (FB)	undirected	4,039	88,234	43.7
Wiki-Vote (WV)	directed	7,115	103,689	29.1
Flixster (FS)	directed	28,843	119,018	9.6
EmailAll (EA)	directed	265,214	420,045	3.2
DBLP (DB)	undirected	317,080	1,049,866	6.6
Twitter (TT)	directed	81,306	1,768,149	43.5
Stanford (SF)	directed	281,903	2,312,497	16.4
Youtube (YT)	undirected	1,134,890	2,987,624	5.3
Pokec (PK)	directed	1,632,803	30,622,564	37.5
Orkut (OK)	undirected	3,072,441	117,185,083	76.3

Leveraging these core results and adopting an analysis similar to Section 6.3, Theorem 7.2 is established. Notably, SA+ provides the same worst-case guarantee as the state-of-the-art solution SA [40], yet *without relying on any assumptions*.

Theorem 7.2. *Given $0 \leq \epsilon, \delta, \leq 1$, SA+ returns S_t^* satisfies:*

$$\Pr[\mu(S_t^*) \geq \kappa(1 - 1/e - \epsilon) \cdot \mu(S_t^o)] \geq 1 - 3\delta. \tag{11}$$

where $\kappa = \max\left\{\frac{\mu_l(S_t^{l,o})}{\mu(S_t^o)}, \frac{\mu(S_t^u)}{\mu_u(S_t^u)}\right\}\frac{1-\epsilon}{1+\epsilon}$ and S_t^o (resp. $S_t^{l,o}$) is the size-k optimal solution for the MM problem (resp. maximizing $\mu_l(\cdot)$).

8 Experiments

This section conducts extensive experiments on real-world graphs to validate the performance of our proposed algorithms. All programs are implemented in C++ and performed on a PC with an Intel Xeon 2.10GHz CPU and 512GB memory. The source code is available at [1].

Algorithms. We implement and evaluate the following algorithms.

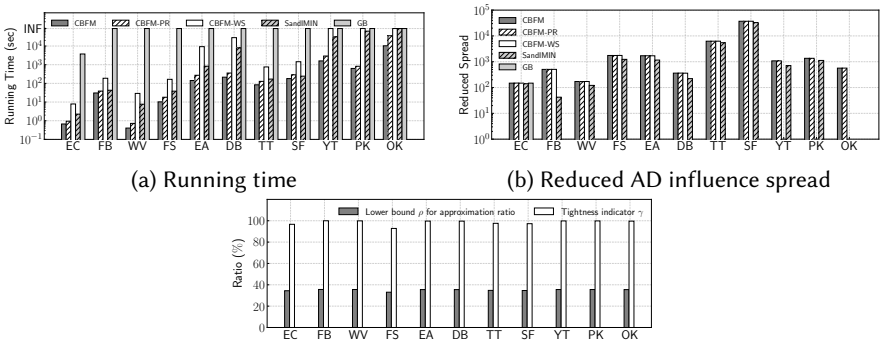
- GB: heuristic baseline proposed in Section 3.3.
- CBFM: Algorithm 5 described in Section 6.
- CBFM-WS: CBFM substitutes WS with US (i.e., uniform sampling) for source node selection of a random sample set.
- CBFM-PR: CBFM without the path-based pruning rule.
- MC [52], GR [59], SandIMIN [51]: existing IMIN algorithms.
- SA [40]: the state-of-the-art MM algorithm.

Datasets. We use 11 real-world datasets in our experiments, with the details reported in Table 2. Flixster[3] is derived from a social movie rating network, where the diffusion probabilities of edges are learned from actual user interaction logs, using the method in [3]. For the other datasets[4], which lack real activation logs, we follow the convention [20, 24, 46–48] and set the diffusion probability $p(u, v)$ of each edge $\langle u, v\rangle$ as the inverse of v's in-degree.

Parameter settings. We set the number of simulations r to 10^4 for GB, and $\epsilon = 0.1, \delta = 1/n$ for the other evaluated algorithms. The default values of $|S|$, k, and T are 10, 20 and 64, respectively. The nodes in the seed set are randomly selected from the top 200 most influential nodes (identified by OPIM-C [46]).

[3]https://www.flixster.com

[4]http://snap.stanford.edu



(a) Running time (b) Reduced AD influence spread

(c) Approximation ratio of CBFM

Fig. 3. Overall performance evaluation under default setting

Delay distribution. Following prior studies [29, 30], we set the delay distribution $\mathcal{D}$ as a Poisson distribution with parameter $\lambda = 2$ by default. Here, λ can be learned from real data and interpreted as the average time it takes for users to revisit the online social network.

We also consider a node-specific setting for $\mathcal{D}$, following [7, 40]. That is, each $u \in V$ is associated with a delay distribution $\mathcal{D}_u$, defined as a Geometric distribution with parameter $p = 5/(|N_{\text{out}}(u)| + 5)$. The intuition is that the more friends u has, the less likely it is for u to meet a certain individual in one timestamp.

Weight function. As discussed before, in the context of the U.S. presidential election, the impact of misinformation on a voter diminishes markedly once his/her vote has been submitted. Additionally, a large proportion of voters tend to cast their ballots within the first few days of the voting period. For example, in Texas's 2020 early voting period, which lasted 18 days, approximately 60% of votes were cast within the first week [13]. These observations suggest that the impact of misinformation often declines rapidly during the initial phase of the voting period. To model this behavior, we adopt the following default setting: $U(T - t_v) = T/(t_v + 1)^A$ when $T - t_v > 0$, where A is a tunable parameter that determines the decay rate of $U(T - t_v)$ with respect to t_v. By default, we set $A = 1$. To model the deadline constraint, we fix $U(T - t_v) = 0$ when $T - t_v \leq 0$.

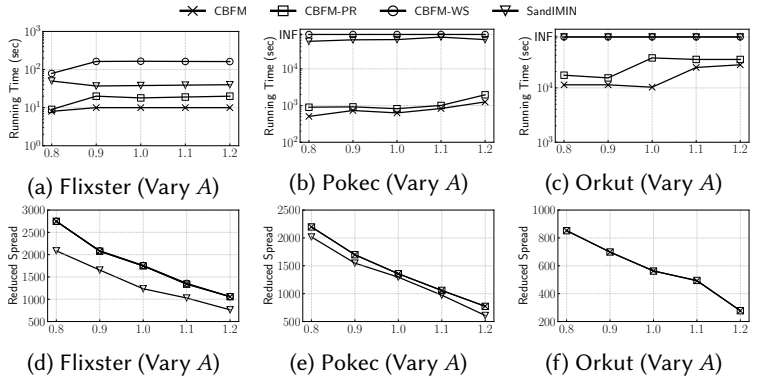
Moreover, we propose a node-specific weight function derivation scheme informed by empirical data. Specifically, for each $v \in V$, based on the average voting time point x_v of each user v (calculated from historical election records), the weight function U_v for v is defined as a piecewise function, i.e.,

$$U_v(T - t_v) = \begin{cases} 1, & 0 \leq t_v \leq \max\{x_v - B, 0\} \\ 1 - \frac{t_v - (x_v - B)}{2B}, & \max\{x_v - B, 0\} < t_v \leq \min\{x_v + B, T\} \\ 0, & t_v > \min\{x_v + B, T\} \end{cases}$$

To account for potential deviations between x_v and the actual voting time point of user v in future elections, here we introduce a tolerance parameter B, which defines a smooth transition interval around x_v. We set $B = 0.1T$ by default. In the experiments, due to the lack of real data, we adopt a stratified sampling strategy to sample each x_v. In particular, 60% of the nodes in V are each assigned a x_v uniformly sampled from the interval $[0, 0.3T]$, and the remaining 40% from the interval $[0.3T, T]$. This schema captures the observation that most users tend to vote during the early days of the election.

For each parameter setting, we repeat each algorithm 10 times and report the average value. We terminate the algorithms that cannot finish within 24 hours and set their results as **INF**.

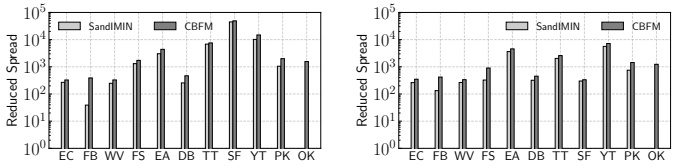
8.1 Experimental Results



(a) Flixster (Vary A) (b) Pokec (Vary A) (c) Orkut (Vary A)

(d) Flixster (Vary A) (e) Pokec (Vary A) (f) Orkut (Vary A)

Fig. 4. Performance evaluation by varying A



(a) HIST (b) DEG

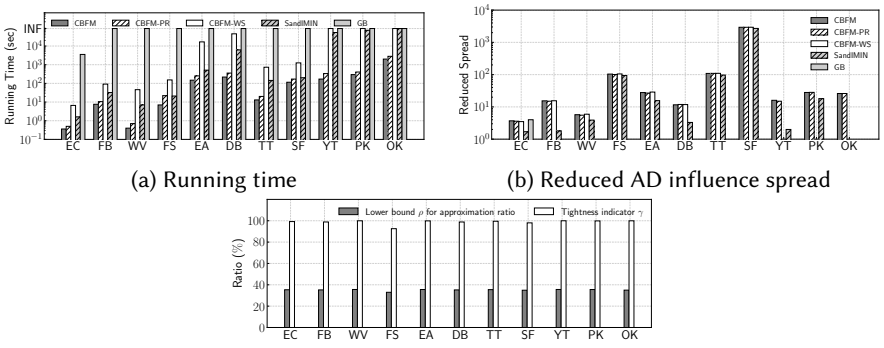
Fig. 5. Effectiveness evaluation by varying seeding approach

Exp-1: Overall performance. We first evaluate the performance of CBFM on all datasets under the default settings. Figure 3(a) first reports the running time. As shown, GB can only complete on the smallest dataset, on which CBFM achieves four orders of magnitude speedup. In addition, CBFM is always more efficient than both CBFM-PR and CBFM-WS. This demonstrates the effectiveness of the pruning rule and WS sampling technique within WS-SSC. CBFM is also consistently faster than SandIMIN, owing to its more advanced sampling approach, as explained in Section 6.4.

Figure 3(b) presents the reduced AD influence spread. As shown, on EC, the only dataset where GB completes, CBFM exhibits comparable performance to GB. This validates the tightness of our proposed bounding functions. Additionally, CBFM performs similarly to CBFM-WS and CBFM-PR, which is not surprising as the techniques excluded in CBFM-WS and CBFM-PR are designed solely to improve efficiency without affecting correctness. Moreover, CBFM always achieves greater spread reduction than SandIMIN, as the latter does not consider the temporal aspects.

We remark that the effectiveness of CBFM will be impacted by the structural properties of the *S-induced activation subgraph*, i.e., a subgraph of the underlying graph that consists of nodes likely to be activated by S before T, and edges with high diffusion probabilities. Generally, CBFM tends to attain greater spread reduction when the activation subgraph exhibits the following properties: (1) a large number of nodes, which provides more opportunities for protection; and (2) a sparse structure, where the removal of a blocker is more likely to delay the activation timestamps of other nodes. For instance, the SF dataset yields the activation subgraph with the largest number of nodes, which is a key factor contributing to the superior spread reduction achieved by CBFM on SF.

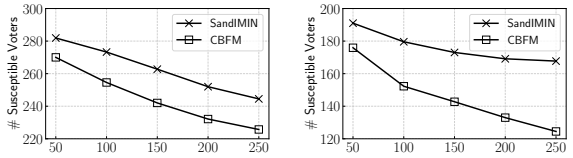
We also assess the approximation guarantees of CBFM. Given that computing CBFM's exact approximation ratio $\tau \cdot (1 - 1/e - \epsilon)$ is #P-hard (see Eq. (8)), here we present its computable lower bound $\rho = \underline{\tau} \cdot (1 - 1/e - \epsilon)$. In particular, let $\hat{\mathbb{D}}_U(\cdot)$ (resp. $\hat{\mathbb{D}}(\cdot)$) be the (ϵ, δ)-estimates of $\mathbb{D}_U(\cdot)$ (resp. $\mathbb{D}(\cdot)$), it then follows that $\underline{\tau} = \frac{\hat{\mathbb{D}}(B_U)(1-\epsilon)^2}{\hat{\mathbb{D}}_U(B_U)(1+\epsilon)^2}$ is a lower bound of τ with at least $1 - 2\delta$ probability. Figure 3(c) reports the value of ρ and $\gamma = \hat{\mathbb{D}}(B_U)/\hat{\mathbb{D}}_U(B_U)$ of CBFM, where γ determines ρ and



(a) Running time (b) Reduced AD influence spread

(c) Approximation ratio of CBFM

Fig. 6. Overall performance evaluation under node-specific setting



(a) Facebook (Vary k) (b) Twitter (Vary k)

Fig. 7. Case study on online social networks

reflects the tightness of upper bounding function. We observe that, ρ consistently exceeds 30%, indicating that CBFM offers strong approximation guarantees. Moreover, γ remains above 90% across all cases, suggesting our bound closely approximates the actual objective value in practice.

Exp-2: Varying A. Next, we evaluate the performance of CBFM by varying A, the parameter in the default setting of $U(\cdot)$. Figure 4 reports the results on FS (with real diffusion probabilities) and the two largest datasets, PK and OK. As shown, the time cost of CBFM remains stable with A increases, reflecting its robustness. Meanwhile, since a larger A leads to a smaller AD influence spread of S, the spread reduction decreases. Additional results under varying $|S|, k, T$ are provided in Appendix A.5 of the full version [1] due to limited space.

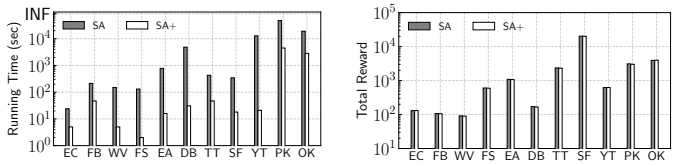
Exp-3: Varying seeding approach. We then evaluate the effectiveness of CBFM when the 200 most influential nodes (from which the seed set S is selected) are identified by HIST [20] and DEG. HIST is an advanced $(1 - 1/e - \epsilon)$-approximation algorithm for IM, while DEG is a heuristic that selects the highest-degree nodes as seeds. Figure 5 reports the results over all the datasets. As can be seen, CBFM always achieves higher spread reduction than SandIMIN. Moreover, CBFM performs comparably under both seeding approaches in most cases.

Exp-4: Results under node-specific settings. We proceed to evaluate CBFM under node-specific settings of both the delay distribution and the weight function. Figure 6 shows the results on all the datasets. The overall trend is consistent with our earlier findings: CBFM dominates SandIMIN in terms of spread reduction, while offering superior efficiency and approximation guarantees.

Exp-5: Case study. Afterwards, we conduct a case study on two online social networks, i.e., Facebook and Twitter. In particular, we consider a presidential election scenario, where 10 influential users release misinformation to interfere with the election. We adopt the node-specific settings for $U(\cdot)$ and $\mathcal{D}$, and set $T = 64$. Moreover, under the assumption that each voter v will cast their ballot around their individual historical average voting time x_v, we sample their actual voting time from a normal distribution $\mathcal{N}(x_v, 0.1T)$.

Table 3. Performance evaluation on IMIN with $k = 100$, $|S| = 10$

	Twitter			Stanford			Youtube			Pokec			Orkut		
	GR	SandIMIN	CBFM	GR	SandIMIN	CBFM	GR	SandIMIN	CBFM	GR	SandIMIN	CBFM	GR	SandIMIN	CBFM
Reduced influence spread	**734.4**	727.3	721.1	**861.9**	855.6	844.2	**1568**	1534	1513	-	1517	**1522**	-	-	**1234**
Approximation ratio (%)	-	28.3	**31.5**	-	32.1	**34.1**	-	32.5	**33.3**	-	31.3	**33.8**	-	-	**32.3**
Running time (s)	4560	432.8	**43.1**	11315	691.3	**39.1**	46312	19970	**1056**	INF	54782	**1179**	INF	INF	**10122**
Memory usage (GB)	**0.11**	3.37	0.14	**0.23**	0.82	0.27	**0.75**	20.86	0.93	-	45.72	**2.56**	-	-	**14.29**



(a) Running time (b) Total reward

Fig. 8. Performance evaluation on MM with $k = 20$, $|S_m| = 10$

We refer to users who have not yet voted but are exposed to misinformation as *susceptible voters*. As previously discussed, misinformation can significantly impact their final voting decisions. Therefore, reducing the number of susceptible voters is critical to preserving electoral fairness. Figure 7 compares the performance of SandIMIN and CBFM in terms of the number of susceptible voters on Facebook and Twitter, as the budget k varies. As we can see, on both networks, CBFM always leads to fewer susceptible voters than SandIMIN. On Twitter, the performance gap between the two methods widens as k increases. These results demonstrate the superior effectiveness of CBFM over SandIMIN, and underscore the practical relevance of addressing the TCIM problem.

Exp-6: Results on the IMIN problem. We then evaluate the performance of CBFM on the IMIN problem, a special case of TCIM, by comparing it with all existing IMIN algorithms, i.e., MC [52], GR [59], SandIMIN [51], using the parameter settings as specified in their original papers. Table 3 shows the results on the five largest datasets; MC fails to complete on any of them, so its results are omitted. Compared to GR and SandIMIN, we draw four key observations. First, CBFM matches their performance in reducing influence spread. Second, CBFM consistently offers a higher approximation ratio. Third, CBFM demonstrates substantial improvements in efficiency. Notably, it is the first algorithm that can handle a graph with 100 million edges. Fourth, in terms of memory usage, CBFM performs comparably with GR and outperforms SandIMIN. In summary, we conclude that CBFM is a theoretically grounded, efficient and effective method with low memory overhead.

Exp-7: Results on the MM problem. We further evaluate our SA+ on the MM problem, compared to the state-of-the-art SA algorithm [40], with parameter settings consistent with those in [40]. Figure 8 shows the results on all the datasets. We find that SA+ attains a comparable reward to SA, while always requiring less running time, with up to two orders of magnitude speedup, due to its advanced stopping condition for sample generation.

9 Conclusion

In this paper, we formulate and study the time-critical influence minimization problem. We devise CBFM, an efficient and effective approximation algorithm for the TCIM problem. As a by-product, we propose the first assumption-free algorithm with approximation guarantee for the misinformation mitigation problem. Extensive experiments show the superiority of our proposed algorithms in terms of effectiveness and efficiency.

Acknowledgments

This work is partially supported by ARC DP240101322, DP230101445, LP210301046, FT210100303. Yanping Wu and Xiaoyang Wang are the corresponding authors.

References

[1] 2025. https://github.com/wjh0116/TCIM.

[2] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives* (2017).

[3] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. 2013. Topic-aware social influence propagation models. *Knowledge and information systems* 37 (2013), 555–584.

[4] Devipsita Bhattacharya and Sudha Ram. 2012. Sharing news articles using 140 characters: A diffusion analysis on Twitter. In *2012 IEEE/ACM International conference on advances in social networks analysis and mining*. IEEE, 966–971.

[5] Andrew An Bian, Joachim M. Buhmann, Andreas Krause, and Sebastian Tschiatschek. 2017. Guarantees for Greedy Maximization of Non-submodular Functions with Applications. In *ICML (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 498–507.

[6] Christian Borgs, Michael Brautbar, Jennifer T. Chayes, and Brendan Lucier. 2014. Maximizing Social Influence in Nearly Optimal Time. In *SODA*. SIAM, 946–957.

[7] Wei Chen, Wei Lu, and Ning Zhang. 2012. Time-Critical Influence Maximization in Social Networks with Time-Delayed Diffusion Process. In *AAAI*. 591–598.

[8] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*. ACM, 1029–1038.

[9] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *KDD*. ACM, 199–208.

[10] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In *ICDM*. IEEE Computer Society, 88–97.

[11] Fan Chung and Linyuan Lu. 2006. Concentration inequalities and martingale inequalities: a survey. *Internet mathematics* 3, 1 (2006), 79–127.

[12] Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F. Werneck. 2014. Timed Influence: Computation and Maximization. *CoRR* abs/1410.6976 (2014).

[13] Kevin DeLuca. 2020. *Spotlight on Texas: Record Breaking Early Vote Turnout.* https://electionlab.mit.edu/articles/spotlight-texas-record-breaking-early-vote-turnout

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

[15] Lidan Fan, Weili Wu, Xuming Zhai, Kai Xing, Wonjun Lee, and Ding-Zhu Du. 2014. Maximizing rumor containment in social networks with constrained time. *Soc. Netw. Anal. Min.* 4, 1 (2014), 214.

[16] Maksym Gabielkov, Arthi Ramachandran, Augustin Chaintreau, and Arnaud Legout. 2016. Social clicks: What and who gets read on Twitter?. In *Proceedings of the 2016 ACM SIGMETRICS international conference on measurement and modeling of computer science*. 179–192.

[17] Arnab Kumar Ghoshal, Nabanita Das, and Soham Das. 2021. Influence of community structure on misinformation containment in online social networks. *Knowl. Based Syst.* 213 (2021), 106693.

[18] Manuel Gomez-Rodriguez, Le Song, Nan Du, Hongyuan Zha, and Bernhard Schölkopf. 2016. Influence Estimation and Maximization in Continuous-Time Diffusion Networks. *ACM Trans. Inf. Syst.* 34, 2 (2016), 9:1–9:33.

[19] Amit Goyal, Wei Lu, and Laks V. S. Lakshmanan. 2011. SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model. In *ICDM*. IEEE Computer Society, 211–220.

[20] Qintian Guo, Sibo Wang, Zhewei Wei, and Ming Chen. 2020. Influence Maximization Revisited: Efficient Reverse Reachable Set Generation with Bound Tightened. In *SIGMOD Conference*. ACM, 2167–2181.

[21] Keke Huang, Sibo Wang, Glenn S. Bevilacqua, Xiaokui Xiao, and Laks V. S. Lakshmanan. 2017. Revisiting the Stop-and-Stare Algorithms for Influence Maximization. *Proc. VLDB Endow.* 10, 9 (2017), 913–924.

[22] José Luis Iribarren and Esteban Moro. 2009. Impact of human activity patterns on the dynamics of information diffusion. *Physical review letters* 103, 3 (2009), 038702.

[23] Márton Karsai, Mikko Kivelä, Raj Kumar Pan, Kimmo Kaski, János Kertész, A-L Barabási, and Jari Saramäki. 2011. Small but slow world: How network topology and burstiness slow down spreading. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 83, 2 (2011), 025102.

[24] David Kempe, Jon M. Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *KDD*. ACM, 137–146.

[25] Arijit Khan. 2016. Towards time-discounted influence maximization. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. 1873–1876.

[26] Thomas Lengauer and Robert Endre Tarjan. 1979. A Fast Algorithm for Finding Dominators in a Flowgraph. *ACM Trans. Program. Lang. Syst.* 1, 1 (1979), 121–141.

[27] Fan Li, Xiaoyang Wang, Dawei Cheng, Cong Chen, Ying Zhang, and Xuemin Lin. 2025. Efficient Dynamic Attributed Graph Generation. In *ICDE*. IEEE, 1415–1428.

[28] Fan Li, Zhiyu Xu, Dawei Cheng, and Xiaoyang Wang. 2024. AdaRisk: Risk-Adaptive Deep Reinforcement Learning for Vulnerable Nodes Detection. *IEEE Trans. Knowl. Data Eng.* 36, 11 (2024), 5576–5590.
[29] Bo Liu, Gao Cong, Dong Xu, and Yifeng Zeng. 2012. Time Constrained Influence Maximization in Social Networks. In *ICDM*. IEEE Computer Society, 439–448.
[30] Bo Liu, Gao Cong, Yifeng Zeng, Dong Xu, and Yeow Meng Chee. 2014. Influence Spreading Path and Its Application to the Time Constrained Social Influence Maximization Problem and Beyond. *IEEE Trans. Knowl. Data Eng.* 26, 8 (2014), 1904–1917.
[31] Wei Lu, Wei Chen, and Laks V. S. Lakshmanan. 2015. From Competition to Complementarity: Comparative Influence Diffusion and Maximization. *Proc. VLDB Endow.* 9, 2 (2015), 60–71.
[32] Mohammad Ali Manouchehri, Mohammad Sadegh Helfroush, and Habibollah Danyali. 2021. Temporal rumor blocking in online social networks: A sampling-based approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52, 7 (2021), 4578–4588.
[33] Amy Mitchell, Galen Stocking, and Katerina Eva Matsa. 2016. Long-form reading shows signs of life in our mobile news world. *Pew Research Center* 5, 05 (2016).
[34] Evgeny Morozov. 2009. Swine flu: Twitter's power to misinform. *Foreign policy* (2009).
[35] Stephen R Neely, Christina Eldredge, Robin Ersing, and Christa Remington. 2022. Vaccine hesitancy and exposure to misinformation: a survey analysis. *Journal of general internal medicine* (2022), 1–9.
[36] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.* 14, 1 (1978), 265–294.
[37] Hung T. Nguyen, My T. Thai, and Thang N. Dinh. 2016. Stop-and-Stare: Optimal Sampling Algorithms for Viral Marketing in Billion-scale Networks. In *SIGMOD Conference*. ACM, 695–710.
[38] Queensland Health. 2024. 2025 Free influenza vaccination program for eligible Queenslanders. https://www.health.qld.gov.au/clinical-practice/guidelines-procedures/diseases-infection/immunisation/service-providers/2025-free-flu-vaccination-program
[39] Revolt. 2020. How to Change Your Vote: All the Details. https://www.revolt.tv/article/how-to-change-your-vote-all-the-details
[40] Michael Simpson, Laks V. S. Lakshmanan, and Farnoosh Hashemi. 2022. Misinformation Mitigation under Differential Propagation Rates and Temporal Penalties. *Proc. VLDB Endow.* 15, 10 (2022), 2216–2229.
[41] Chonggang Song, Wynne Hsu, and Mong Li Lee. 2017. Temporal influence blocking: Minimizing the effect of misinformation in social networks. In *2017 IEEE 33rd international conference on data engineering (ICDE)*. IEEE, 847–858.
[42] Lichao Sun, Xiaobin Rui, and Wei Chen. 2023. Scalable Adversarial Attack Algorithms on Influence Maximization. In *WSDM*. ACM, 760–768.
[43] Xingyu Tan, Jingya Qian, Chen Chen, Sima Qing, Yanping Wu, Xiaoyang Wang, and Wenjie Zhang. 2023. Higher-Order Peak Decomposition. In *CIKM*. ACM, 4310–4314.
[44] Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, and Wenjie Zhang. 2025. Paths-over-Graph: Knowledge Graph Empowered Large Language Model Reasoning. In *WWW*. ACM, 3505–3522.
[45] Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, Liming Zhu, and Wenjie Zhang. 2025. Hydra: Structured Cross-Source Enhanced Large Language Model Reasoning. *CoRR* abs/2505.17464 (2025).
[46] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online Processing Algorithms for Influence Maximization. In *SIGMOD Conference*. ACM, 991–1005.
[47] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence Maximization in Near-Linear Time: A Martingale Approach. In *SIGMOD Conference*. ACM, 1539–1554.
[48] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: near-optimal time complexity meets practical efficiency. In *SIGMOD Conference*. ACM, 75–86.
[49] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *science* 359, 6380 (2018), 1146–1151.
[50] Jinghao Wang, Yanping Wu, Xiaoyang Wang, Chen Chen, Ying Zhang, and Lu Qin. 2025. Effective Influence Maximization with Priority. In *WWW*. ACM, 4673–4683.
[51] Jinghao Wang, Yanping Wu, Xiaoyang Wang, Ying Zhang, Lu Qin, Wenjie Zhang, and Xuemin Lin. 2024. Efficient Influence Minimization via Node Blocking. *Proc. VLDB Endow.* 17, 10 (2024), 2501–2513.
[52] Senzhang Wang, Xiaojian Zhao, Yan Chen, Zhoujun Li, Kai Zhang, and Jiali Xia. 2013. Negative Influence Minimizing by Blocking Nodes in Social Networks. In *AAAI (Late-Breaking Developments) (AAAI Technical Report, Vol. WS-13-17)*. AAAI.
[53] Xiaoyang Wang, Ying Zhang, Wenjie Zhang, Xuemin Lin, and Chen Chen. 2017. Bring Order into the Samples: A Novel Scalable Method for Influence Maximization. *IEEE Trans. Knowl. Data Eng.* 29, 2 (2017), 243–256.

[54] Darrell M. West. 2024. How disinformation defined the 2024 election narrative. *Brookings Institution* (November 2024). https://www.brookings.edu/articles/how-disinformation-defined-the-2024-election-narrative/
[55] Yanping Wu, Renjie Sun, Xiaoyang Wang, Dong Wen, Ying Zhang, Lu Qin, and Xuemin Lin. 2024. Efficient Maximal Frequent Group Enumeration in Temporal Bipartite Graphs. *Proc. VLDB Endow.* 17, 11 (2024), 3243–3255.
[56] Yanping Wu, Renjie Sun, Xiaoyang Wang, Ying Zhang, Lu Qin, Wenjie Zhang, and Xuemin Lin. 2024. Efficient Maximal Temporal Plex Enumeration. In *ICDE*. IEEE, 3098–3110.
[57] Yanping Wu, Jinghao Wang, Renjie Sun, Chen Chen, Xiaoyang Wang, and Ying Zhang. 2024. Targeted Filter Bubbles Mitigating via Edges Insertion. In *WWW (Companion Volume)*. ACM, 734–737.
[58] Fangsong Xiang, Jinghao Wang, Yanping Wu, Xiaoyang Wang, Chen Chen, and Ying Zhang. 2024. Rumor blocking with pertinence set in large graphs. *World Wide Web (WWW)* 27, 1 (2024), 6.
[59] Jiadong Xie, Fan Zhang, Kai Wang, Xuemin Lin, and Wenjie Zhang. 2023. Minimizing the Influence of Misinformation via Vertex Blocking. In *ICDE*. IEEE, 789–801.
[60] Jiadong Xie, Fan Zhang, Kai Wang, Jialu Liu, Xuemin Lin, and Wenjie Zhang. 2025. Influence minimization via blocking strategies. *INFORMS Journal on Computing* (2025).
[61] Lan Yang and Zhiwu Li. 2024. Deadline-aware misinformation prevention in social networks with time-decaying influence. *Expert Systems with Applications* 238 (2024), 121847.
[62] Zian Zhai, Fan Li, Xingyu Tan, Xiaoyang Wang, and Wenjie Zhang. 2025. Graph is a Natural Regularization: Revisiting Vector Quantization for Graph Representation Learning. *CoRR* abs/2508.06588 (2025).
[63] Zian Zhai, Sima Qing, Xiaoyang Wang, and Wenjie Zhang. 2024. Adapting Unsigned Graph Neural Networks for Signed Graphs: A Few-Shot Prompt Tuning Approach. *CoRR* abs/2412.12155 (2024).
[64] Yuqing Zhu, Jing Tang, Xueyan Tang, Sibo Wang, and Andrew Lim. 2023. 2-hop+ Sampling: Efficient and Effective Influence Estimation. *IEEE Trans. Knowl. Data Eng.* 35, 2 (2023), 1088–1103.

Received April 2025; revised July 2025; accepted August 2025