Approximating Contribution in Misinformation Mitigation

JINGHAO WANG, University of Technology Sydney, Australia
RENZHE MIAO, Zhejiang Gongshang University, China
YANPING WU, University of Technology Sydney, Australia
XIAOYANG WANG, The University of New South Wales, Australia
YING ZHANG, University of Technology Sydney, Australia

Online social networks (OSNs) have rapidly emerged as prominent platforms for sharing diverse content, where the word-of-mouth effect drives extensive information propagation. However, such powerful dissemination capabilities also accelerate the spread of misinformation, resulting in significant negative impacts. To combat this, the OSN provider often incentivizes some influencers to initiate the truth campaign, enabling users to receive truth before misinformation and thus suppressing its further spread. This mitigation strategy has proven effective in practice and has consequently attracted substantial research interest, particularly in optimizing influencer selection. In this paper, complementing the existing performance-centric line of work, we address the fair allocation of incentives for engaged influencers, an underexplored dimension that is critical for ensuring the long-term sustainability of misinformation mitigation. We first define a novel contribution metric and reveal its rationality by connecting it to the Shapley value, a classic fairness concept in cooperative game theory. We then formulate the *mitigation contribution allocation* (MCA) problem, which seeks to compute, for each influencer, the expected mitigation contribution conditioned on the observed mitigation result (which corresponds to exponentially many possible diffusion cascades). In view of the prohibitive cost of exactly solving MCA via exhaustive enumeration, we propose VG-App, an efficient sampling-based approximation algorithm with provable accuracy guarantees. VG-App is built upon the advanced VG-Sampling method, which reduces the global sampling task to independent node-wise decisions, drastically pruning the sample space to ensure that generated samples are consistent with the observed mitigation result, while maintaining unbiasedness. Extensive empirical studies on large real-world datasets demonstrate that VG-App achieves accurate approximations while preserving high efficiency.

CCS Concepts: • **Theory of computation → Social networks**.

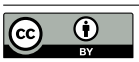
Additional Key Words and Phrases: Misinformation Mitigation; Approximation Algorithm

ACM Reference Format:
Jinghao Wang, Renzhe Miao, Yanping Wu, Xiaoyang Wang, and Ying Zhang. 2026. Approximating Contribution in Misinformation Mitigation. *Proc. ACM Manag. Data* 4, 4 (SIGMOD), Article 258 (September 2026), 27 pages. https://doi.org/10.1145/3837096

1 Introduction

Online social networks (OSNs), such as Facebook and X, have rapidly emerged in recent years, where users actively engage in sharing perspectives, ideas, and news. Driven by the word-of-mouth effect, such information can be extensively propagated, making OSN a prominent medium for

Authors' Contact Information: Jinghao Wang, University of Technology Sydney, Sydney, Australia, jinghao.wang@student.uts.edu.au; Renzhe Miao, Zhejiang Gongshang University, Hangzhou, China, 23020100017@pop.zjgsu.edu.cn; 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.



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 2836-6573/2026/9-ART258
https://doi.org/10.1145/3837096

information diffusion [3, 9, 20]. However, the massive user base and rapid dissemination capabilities of OSNs also accelerate the spread of misinformation, resulting in significant adverse consequences. For instance, misinformation about public health crises can damage public trust, prevent individuals from adopting safety measures, and cause serious harm to society [39].

To mitigate misinformation, a widely adopted approach for OSN providers is to incentivize influencers to disseminate truthful information, enabling users to receive the truth before misinformation, thereby suppressing its further spread. In general, this mitigation process unfolds in two stages. Initially, upon detecting the source of misinformation, the OSN provider identifies a set of influencers, with the goal of maximizing the overall mitigation impact. Then, the provider invites these influencers to participate in the mitigation campaign, and those who accept the invitation execute the campaign and are rewarded accordingly [17, 19, 35].

Existing research has primarily focused on the first stage, developing efficient algorithms for influencer selection under cardinality constraints [12, 41, 42, 44, 52–54]. In contrast, the subsequent stage, which governs how incentives are allocated among engaged influencers, remains unaddressed in the literature, despite being equally indispensable to the mitigation pipeline. In particular, a rewarding scheme that fails to fairly reflect each influencer's achieved contribution will undermine their willingness to participate in future mitigation campaigns [2], ultimately compromising the long-term sustainability of misinformation mitigation in OSNs.

To bridge this gap, this work initiates the study of fair incentive allocation in misinformation mitigation. We adopt a result-based allocation scheme, where each engaged influencer is compensated, after the campaign concludes, according to their individual contribution to the observed mitigation result. This choice is well-motivated on two grounds. First, it reflects a common practice in incentive allocation. For example, in online advertising, result-based payment is the norm, with advertisers paying based on actual clicks or conversions [4]. Second, it benefits both parties involved: from the influencers' perspective, it mirrors the logic of performance-based bonuses, a standard and widely accepted compensation mechanism in the workplace; from the provider's perspective, tying rewards to actual performance encourages influencers to genuinely commit effort to the mitigation campaign [31, 38].

Realizing such a scheme requires a reliable method for quantifying each influencer's individual contribution under a given mitigation result. An intuitive strategy that allocates contributions based on the mitigation achieved by each influencer may be inaccurate, as multiple influencers can simultaneously trigger the mitigation for the same users. To resolve such overlap, we propose *mitigation contribution* (Definition 2.3), a principled metric that, for each user benefiting from the mitigation, distributes the credit equally among all contributing influencers. By establishing its equivalence to the Shapley value (Lemma 2.5), a classic solution concept in cooperative game theory, our contribution metric is guaranteed to satisfy the well-established fairness axioms [40], thereby confirming its theoretical soundness.

Computing the mitigation contribution in practice, however, is non-trivial. To elaborate, it hinges on underlying diffusion cascade, which includes all edge-level successful information propagations within the mitigation campaign. In real-world OSNs, such granular data is rarely accessible, since user interaction logs are typically restricted and protected by privacy regulations [18]. Instead, the activation statuses of users (i.e., whether they share truth, misinformation, or neither) are commonly observable, as such actions are directly visible on the platform. This motivates us to operate under a practical node-level feedback setting, in which only the activation statuses of users are revealed after a mitigation campaign, while the edge-level diffusion outcomes remain unobserved. Within this setting, we formulate the *mitigation contribution allocation* (MCA) problem. Given a directed graph G where information diffusion follows the competitive independent cascade model [12], a set S_M of seeds initiating the misinformation, a set S_T of seeds initiating the truth

with $S_T \cap S_M = \emptyset$, and a node-level observation $\mathcal{T}$, the goal of MCA is to compute, for each truth seed $s \in S_T$, the expected mitigation contribution over all valid cascades. Here, a cascade stemming from S_T and S_M is valid if it yields node statuses consistent with the given observation $\mathcal{T}$.

Challenges. Exactly solving the MCA problem by enumerating all valid cascades is computationally intractable, as each observation $\mathcal{T}$ corresponds to exponentially many potential cascades. In view of this, we aim to efficiently approximate the exact solution within an acceptable error through *sampling*, which, nevertheless, faces the following two main challenges.

- *Challenge I: Unbiased sampling of valid cascades.* A straightforward sampling method, which independently specifies node and edge statuses within the underlying graph, would result in a vast sample space containing only an extreme sparsity of valid samples. Given that invalid samples fail to yield the observation $\mathcal{T}$ and thus do not contribute to the approximation, it is essential to develop a sampling mechanism that not only avoids invalid samples but also ensures unbiased results.
- *Challenge II: Determining an appropriate sample size.* The law of large numbers implies that an accurate approximation requires a sufficient number of samples; however, excessive sampling inevitably incurs substantial computational overhead. Therefore, it is critical to eliminate redundant samples and strike a delicate balance between accuracy and efficiency.

Our solution. Guided by these challenges, we first propose the VG-Sampling approach for cascade sampling, which strictly enforces the validity of each sampled cascade. We start by establishing the necessary and sufficient condition for generating a valid cascade (Lemma 3.4), a key result that decomposes the complex global sampling task into a collection of independent edge selection problems. Leveraging this insight, VG-Sampling runs in a node-wise manner, employing a partition-based edge selection strategy to inherently bypass the generation of invalid cascades, thereby drastically pruning the original sample space induced by straightforward sampling. Theoretically, we prove that VG-Sampling yields unbiased samples: for any two cascades, the ratio of their generation probabilities under VG-Sampling exactly matches that in the original sample space (Lemma 3.5). We also derive a factor that quantifies the compression ratio between the original sample space and that under VG-Sampling (Lemma 3.6). In practice, this factor can be astronomically large; for instance, it reaches $10^{25023.7}$ on the Orkut dataset (Table 4), demonstrating the strong effectiveness of our technique. Beyond its advanced cascade-sampling capability, VG-Sampling further supports the computation of mitigation contributions for each seed under the generated cascade with negligible computational cost, by maintaining lightweight auxiliary information during the sampling process.

Building upon VG-Sampling, we develop VG-App, an accurate and efficient approximation algorithm that, for any seed $s \in S_T$, returns an estimate of its expected mitigation contribution within a multiplicative factor of $1 \pm \epsilon$ with probability at least $1 - \delta$ (Theorem 4.1), where ϵ and δ are user-specified error parameters. At a high level, VG-App employs an adaptive sampling schema that incrementally generates samples via VG-Sampling, until all seed-specific stopping conditions are met. Powered by the tightness of the derived upper bound on each seed's mitigation contribution across all valid cascades, the stopping conditions are able to significantly reduce the required sample size without compromising the approximation guarantees (Table 5).

Extensive experiments on 5 real-world datasets are conducted to verify the performance of our proposed algorithms. In general, VG-App achieves high accuracy, with the mean squared error of its outputs being several orders of magnitude lower than that of heuristic baselines. Moreover, VG-App demonstrates superior scalability: it efficiently processes the largest dataset containing over 100 million edges, whereas the approximation baseline fails to scale to the dataset with only tens of thousands of edges.

Table 1. Frequently used notations

Notation	Description
$G = (V, E)$	the directed graph with node set V and edge set E
M, T	the misinformation and truth campaigns
S_M, S_T	the seed sets for initiating the campaigns M and T
ω	a cascade under the CIC model
$R_\omega(S_M)$	the non-seed nodes activated by M in the absence of T (i.e., when $S_T = \emptyset$) in ω
$\Psi_\omega(s)$	the nodes that can receive T from s in ω
$M_\omega(v)$	the seeds in S_T enabling v to receive T in ω
$\rho_\omega^v(S_T)$	the mitigation effect achieved by S_T on node v in ω
$\sigma_\omega(s)$	the mitigation contribution of seed s in ω
$\mathcal{T}, G[\mathcal{T}]$	the observation, the observation-driven graph
$\Psi(S_M), \Psi(S_T)$	the set of non-seed nodes activated by the campaigns M and T, as given in $\mathcal{T}$
$\mathbb{E}_\mathcal{T}[\sigma(s)]$	the expectation of $\sigma_\omega(s)$ conditioned on $\mathcal{T}$
$g, \omega(g)$	the possible world of $G[\mathcal{T}]$, the corresponding cascade

Contributions. The key contributions are summarized as follows.

- We define the mitigation contribution and confirm its rationality by establishing a direct connection to the Shapley value.
- We formulate the MCA problem aiming to compute the expected mitigation contribution conditioned on the given observation $\mathcal{T}$.
- We design VG-Sampling, an advanced method for cascade sampling that guarantees the validity of every produced cascade.
- We devise VG-App, an efficient approximation algorithm with provable accuracy guarantees for the MCA problem.
- We conduct extensive experiments on real-world datasets to show the superior performance of our proposed approaches.

Roadmap. The remainder of this paper is organized below. Section 2 introduces the contribution metric, relates it to the Shapley value, and formulates the MCA problem. Section 3 details the VG-Sampling approach, while Section 4 presents the VG-App algorithm. Extensive experimental results are shown in Section 5. Section 6 reviews the related work, and Section 7 concludes the paper.

2 Preliminaries

In this paper, we model an online social network as 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. Each edge $\langle u, v\rangle \in E$ is associated with a diffusion probability $p(u, v) \in [0, 1]$. For an edge $\langle u, v\rangle$, we refer to u as an incoming neighbor of v, v as an outgoing neighbor of u, and $\langle u, v\rangle$ as an incoming edge of v. Table 1 lists the notations frequently used throughout the paper. Section 2.1 introduces the diffusion model and the mitigation metric. Section 2.2 then details the formulation of mitigation contribution, while Section 2.3 establishes its theoretical equivalence to the Shapley value. We present the formal problem statement in Section 2.4.

2.1 Competitive Information Diffusion

Diffusion model. We employ the *competitive independent cascade* (CIC) model [12]. Let M denote the *misinformation* campaign, and T denote the *truth* campaign. Given seed sets S_M and S_T that

initiate campaigns M and T respectively, the diffusion process under the CIC model proceeds in discrete time steps, as described below.

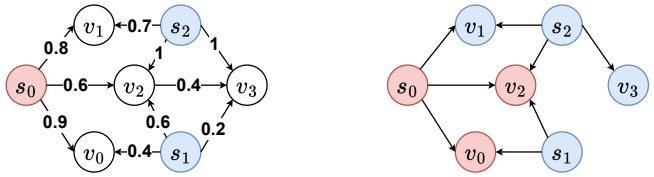
- At timestamp 0, the nodes in S_M (resp. S_T) are activated by the campaign M (resp. T). Each activated node remains active in all subsequent timestamps, with its associated campaign unchanged.
- If a node u is activated by the M (resp. T) campaign at timestamp t, then at timestamp $t+1$ it has a single opportunity to enable each of its *inactive* outgoing neighbors v to receive M (resp. T), with a success probability of $p(u,v)$.
- A node v that receives only one campaign is activated by that campaign. Otherwise, if both campaigns are received simultaneously, a *tie-breaking* policy is applied: node v is activated by M with probability p_M^v and by T with probability p_T^v, where $p_M^v, p_T^v \in [0,1]$ are node-specific parameters such that $p_M^v + p_T^v = 1$. Note that receiving a campaign and being activated by it occur at the same timestamp.
- The propagation process stops when no more nodes can be activated in the graph G.

As a generalization of the classical IC model [28], CIC inherits its proven modeling paradigm and thus provides a principled framework for modeling the fundamental dynamics of competing campaign propagation. In particular, the two-campaign formulation of CIC is a natural and reasonable abstraction: although users may encounter multiple conflicting statements in practice, the truth on a topic is typically unique, while the various non-truthful variants can be collectively treated as a single misinformation campaign. In addition, CIC reflects a realistic diffusion mechanism in which users propagate information with a certain probability upon exposure, with diffusion probabilities that can be learned from empirical action logs [8]. Moreover, the node-specific tie-breaking probabilities p_M^v, p_T^v can also be flexibly specified; for example, p_T^v (resp. p_M^v) can be set as the fraction of the sum of v's incoming edge probabilities that comes from T-active (resp. M-active) neighbors, or learned from each user's historical sharing behavior to encode per-user predispositions toward each campaign. These advantages have driven the wide adoption of CIC in the misinformation mitigation literature [12, 13, 43, 52, 54].

A random CIC diffusion result gives rise to a *cascade* ω, which can be viewed as a subgraph of G, containing the set of activated nodes associated with their activation campaign, and the edges that correspond to successful propagation events. It is worth noting that, since an active node attempts to propagate information only to its inactive outgoing neighbors under the CIC model, every edge $\langle u,v\rangle$ in the cascade ω is constrained by $t(u) = t(v) - 1$, where $t(\cdot)$ denotes the activation timestamp of a node.

Example 2.1. Consider a directed graph G in Figure 1(a), where each edge is labeled with its propagation probability. $S_M = \{s_0\}$ initiates campaign M, while $S_T = \{s_1, s_2\}$ initiates campaign T. Figure 1(b) illustrates a random cascade ω of G, where nodes activated by M are colored red and those by T are blue. It can be observed that nodes v_0, v_1, and v_2 receive both campaigns simultaneously, necessitating a tie-breaking policy to determine their activation statuses. In contrast, node v_3 receives only campaign T and is thus directly activated by it.

Mitigation metric. A natural starting point for evaluating mitigation efficacy is the eventual number of nodes saved from the misinformation campaign. This binary view, however, overlooks an important intermediate state: a user who is activated by misinformation may also be exposed to truthful information, which can partially diminish the negative impact of misinformation. To reflect this, we first introduce a fine-grained mitigation metric for each node, denoted by $\rho_\omega^v(\cdot)$,



(a) Graph G (b) Random cascade ω

Fig. 1. Example of the mitigation contribution

which assigns differentiated rewards based on a user's exposure status under a given cascade ω.

$$\rho_\omega^v(S_T) = \begin{cases} \Delta & \text{if } v \text{ is activated by } T, \\ \Lambda & \text{if } v \text{ is activated by } M \text{ via tie-breaking}, \\ 0 & \text{if } v \text{ is activated by } M \text{ without tie-breaking}, \end{cases} \tag{1}$$

where Δ and Λ are predefined parameters satisfying $0 \le \Lambda < \Delta$. Specifically, the metric captures different levels of mitigation: Δ rewards the best outcome, where a user shares the truth; Λ rewards a suboptimal outcome, where a user shares the misinformation but has also received the truth; and 0 corresponds to the worst case, where a user receives only misinformation. These parameters are flexible and tunable based on the application scenario. For example, in financial-misinformation mitigation, even when a user shares a misleading investment claim, awareness of the truth may still protect the user personally by encouraging more cautious investment decisions; setting Λ to a positive value thus allows the metric to capture this partial mitigation effect.

With this in place, we then define the overall mitigation achieved by the truth seed set S_T in ω as $\rho_\omega(S_T) = \sum_{v \in R_\omega(S_M)} \rho_\omega^v(S_T)$. Here we restrict the analysis to $R_\omega(S_M)$, which is the set of non-seed nodes activated by M in the absence of T (i.e., when $S_T = \emptyset$) under ω, as only the nodes activated by misinformation could benefit from the mitigation effect.

Example 2.2. Consider the cascade ω in Figure 1(b), where $S_M = \{s_0\}$ and $S_T = \{s_1, s_2\}$. We have $\rho_\omega(S_T) = \rho_\omega^{v_0}(S_T) + \rho_\omega^{v_1}(S_T) + \rho_\omega^{v_2}(S_T) = \Lambda + \Delta + \Lambda = 2\Lambda + \Delta$. Notably, although v_3 is activated by T, the mitigation effect of S_T on v_3 is 0, since $v_3 \notin R_\omega(S_M)$.

2.2 Mitigation Contribution

In this paper, we seek to analyze the individual contribution of each seed $s \in S_T$ in the misinformation mitigation campaign. Given a specific cascade ω, one might intuitively regard the mitigation $\rho_\omega(s)$ as an indicator of each seed's contribution. However, this measure is generally inaccurate since multiple seeds in S_T may simultaneously trigger the mitigation of a node v (i.e., enable v to receive the truth). For example, in Figure 1(b), seeds s_1 and s_2 mitigate node v_2 at the same time, leading to non-additivity where $\rho_\omega(S_T) \neq \sum_{s \in S_T} \rho_\omega(s)$. In view of this, we adopt an equal-share strategy to handle the mitigation overlap between truth seeds: we distribute the credit for a node's mitigation uniformly among all truth seeds that concurrently induce the effect. This allocation rule follows naturally from the independence of propagation across different neighbors. When multiple mitigators reach a user simultaneously, the model offers no means of distinguishing which message drives the user's activation. In such cases, distributing credit evenly among these mitigators is a neutral, symmetry-based allocation that avoids arbitrary bias toward any particular participant. It is also reasonable in real-world scenarios where a user is simultaneously exposed to the same truth from multiple unfamiliar influencers; the user typically has no clear way to judge which source had greater impact.

For each truth seed $s \in S_T$, let $\Psi_\omega(s)$ be the set of nodes that can receive T from s in ω. For each node $v \in R_\omega(S_M)$ that may benefit from mitigation, define $M_\omega(v)$ as the set of seeds in S_T that can enable v to receive T in ω. The formal definition of our contribution metric, termed *mitigation contribution*, is given as follows.

Definition 2.3 (Mitigation Contribution). Given a truth seed set S_T, a misinformation seed set S_M, and a cascade ω initiated by S_T and S_M under the CIC model, the mitigation contribution of each seed $s \in S_T$ in ω, denoted by $\sigma_\omega(s)$, is formulated as

$$\sigma_\omega(s) = \sum_{v \in R_\omega(S_M) \cap \Psi_\omega(s)} \frac{\rho_\omega^v(S_T)}{|M_\omega(v)|}. \tag{2}$$

Example 2.4. Reconsider Figure 1, we proceed to elaborate on the computation of $\sigma_\omega(s_1)$. First, we have $R_\omega(S_M) \cap \Psi_\omega(s_1) = \Psi_\omega(s_1) = \{v_0, v_2\}$. Second, we have $M_\omega(v_0) = \{s_1\}$ and $M_\omega(v_2) = \{s_1, s_2\}$. It then follows that $\sigma_\omega(s_1) = \frac{\rho_\omega^{v_0}(S_T)}{|M_\omega(v_0)|} + \frac{\rho_\omega^{v_2}(S_T)}{|M_\omega(v_2)|} = \frac{\Lambda}{1} + \frac{\Lambda}{2} = \frac{3\Lambda}{2}$.

2.3 Connection to Shapley Value

This section validates the rationality of our contribution metric by establishing a direct connection to the Shapley value.

Originally introduced by Lloyd Shapley [40], the Shapley value serves as a fundamental solution concept in cooperative game theory that equitably allocates the total gains (or costs) among participants, ensuring that each player's payoff reflects their individual contribution to the collective outcome. Specifically, consider a cooperative game (S, f), where S is a set of players and $f : 2^S \to \mathbb{R}_+$ is a characteristic function assigning a value to each coalition of players. The Shapley value of a player $s \in S$ is determined by averaging its marginal contributions across all possible orders in which players may join the coalition. Let Π be the set of all permutations of S, and π_s be the set of players that precede s in permutation π. Formally, the Shapley value of s is defined as

$$\phi_f(s) = \frac{1}{|S|!} \sum_{\pi \in \Pi} \big(f(\pi_s \cup \{s\}) - f(\pi_s)\big).$$

The Shapley value satisfies the properties of *efficiency*, *symmetry*, *linearity*, and *null-player* (see [40] for details), making it a widely accepted notion of fairness in cooperative game theory.

Equivalence under CIC. Generalizing the total mitigation $\rho_\omega(S_T)$ to arbitrary subsets $A \subseteq S_T$, we define the function ρ_ω as $\rho_\omega(A) = \sum_{v \in R_\omega(S_M)} \rho_\omega^v(S_T) \cdot \mathbf{1}_{\{A \cap M_\omega(v) \neq \emptyset\}}$, where the indicator $\mathbf{1}_{\{A \cap M_\omega(v) \neq \emptyset\}}$ equals 1 if $A \cap M_\omega(v) \neq \emptyset$ and 0 otherwise. Let $\phi_{\rho_\omega}(s)$ denote the Shapley value of a seed $s \in S_T$ in the cooperative game (S_T, ρ_ω). The following lemma establishes the equivalence between our mitigation contribution metric and the Shapley value. The intuition is that, in the Shapley-value average over permutations of S_T, a seed s contributes $\rho_\omega^v(S_T)$ at a node $v \in R_\omega(S_M)$ only when it appears first among $M_\omega(v)$ in the permutation order, which happens in a $1/|M_\omega(v)|$ fraction of all permutations, matching the equal-share rule underlying our contribution metric.

Lemma 2.5. *For any given cascade ω and each truth seed $s \in S_T$, it holds that $\phi_{\rho_\omega}(s) = \sigma_\omega(s)$.*

Proof. Let Π_T be the set of all permutations of S_T. By the definition of $\phi_{\rho_\omega}(s)$, we have

$$\begin{aligned} \phi_{\rho_\omega}(s) &= \frac{1}{|S_T|!} \sum_{\pi \in \Pi_T} \big(\rho_\omega(\pi_s \cup \{s\}) - \rho_\omega(\pi_s)\big) \\ &= \frac{1}{|S_T|!} \sum_{\pi \in \Pi_T} \sum_{v \in R_\omega(S_M)} \big(\rho_\omega^v(\pi_s \cup \{s\}) - \rho_\omega^v(\pi_s)\big) \end{aligned} \tag{3}$$

For any $\pi \in \Pi_T$ and $v \in R_\omega(S_M)$, there are three possible cases:
Case 1: $v \notin \Psi_\omega(s)$. In this case, node v is not reachable by the truth from s. Consequently, adding s to the set π_s yields no additional mitigation effect on v, i.e., $\rho_\omega^v(\pi_s \cup \{s\}) - \rho_\omega^v(\pi_s) = 0$.
Case 2: $v \in \Psi_\omega(s)$ and $\pi_s \cap M_\omega(v) \neq \emptyset$. In this case, node v is reachable by the truth from both seed s and the set π_s. This implies that v is also reachable by the truth from $\pi_s \cup \{s\}$. Hence, we have $\rho_\omega^v(\pi_s \cup \{s\}) - \rho_\omega^v(\pi_s) = 0$.
Case 3: $v \in \Psi_\omega(s)$ and $\pi_s \cap M_\omega(v) = \emptyset$. In this case, node v is reachable by the truth from seed s, but not from the set π_s. Thus, the inclusion of s in π_s would trigger the mitigation on node v, i.e., $\rho_\omega^v(\pi_s \cup \{s\}) - \rho_\omega^v(\pi_s) = \rho_\omega^v(S_T)$.

It is evident that Case 3 is the only situation in which the inclusion of s in π_s results in a positive marginal gain of $\rho_\omega^v(S_T)$. Let $\mathbf{1}_{\{s=\text{first}(M_\omega(v))\}}$ be an indicator variable which equals 1 if s is placed at the first position in $M_\omega(v)$ under a permutation π, and 0 otherwise. It can be verified that $\pi_s \cap M_\omega(v) = \emptyset$ and $s \in M_\omega(v)$ (i.e., Case 3) holds if and only if $\mathbf{1}_{\{s=\text{first}(M_\omega(v))\}} = 1$. Combining the above analysis, we can derive that

$$\begin{aligned}
&\sum_{\pi \in \Pi_T} \sum_{v \in R_\omega(S_M)} \left(\rho_\omega^v(\pi_s \cup \{s\}) - \rho_\omega^v(\pi_s)\right) \\
&= \sum_{\pi \in \Pi_T} \sum_{v \in R_\omega(S_M) \cap \Psi_\omega(s)} \rho_\omega^v(S_T) \cdot \mathbf{1}_{\{s=\text{first}(M_\omega(v))\}} \\
&= \sum_{v \in R_\omega(S_M) \cap \Psi_\omega(s)} \rho_\omega^v(S_T) \cdot \frac{|S_T|!}{|M_\omega(v)|}.
\end{aligned}$$

The second equality holds because, among the set Π_T of all $|S_T|!$ permutations, a fraction of $\frac{1}{|M_\omega(v)|}$ satisfies $\mathbf{1}_{\{s=\text{first}(M_\omega(v))\}} = 1$. Plugging this result into Eq. (3), we have

$$\begin{aligned}
\phi_{\rho_\omega}(s) &= \frac{1}{|S_T|!} \sum_{v \in R_\omega(S_M) \cap \Psi_\omega(s)} \rho_\omega^v(S_T) \cdot \frac{|S_T|!}{|M_\omega(v)|} \\
&= \sum_{v \in R_\omega(S_M) \cap \Psi_\omega(s)} \rho_\omega^v(S_T) \cdot \frac{1}{|M_\omega(v)|} = \sigma_\omega(s).
\end{aligned}$$

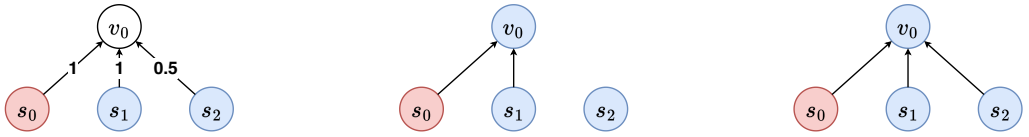
This completes the proof. □

This result confirms that our contribution metric inherits the desirable properties of the Shapley value, thus ensuring theoretical soundness and principled mitigation credit allocation.

Equivalence beyond CIC. The equivalence is in fact not specific to the CIC model. Intuitively, since the proof of Lemma 2.5 touches the diffusion model only through the reachable set $R_\omega(S_M)$, the seed set $M_\omega(v)$, and mitigation value $\rho_\omega^v(S_T)$, the identity should extend to any two-campaign diffusion model whose cascades uniquely determine these quantities, as formalized below.

Lemma 2.6. *Given seed sets S_M and S_T, let $\mathcal{M}$ be a two-campaign diffusion model in which every cascade ω uniquely determines $R_\omega(S_M)$, and further uniquely determines $M_\omega(v)$ and $\rho_\omega^v(S_T)$ for each $v \in R_\omega(S_M)$. Then, for each cascade ω of $\mathcal{M}$ and each truth seed $s \in S_T$, $\phi_{\rho_\omega}(s) = \sigma_\omega(s)$ holds.*

Proof. It can be observed that the derivation of Lemma 2.5 uses the cascade ω only to determine the set $R_\omega(S_M)$, and for each $v \in R_\omega(S_M)$, the set $M_\omega(v)$ and the mitigation value $\rho_\omega^v(S_T)$. These quantities fully specify the characteristic function ρ_ω; the remainder of the derivation, i.e., averaging per-node marginal gains over permutations of S_T, is purely combinatorial and independent of the diffusion dynamics. As every cascade ω of $\mathcal{M}$ uniquely determines $R_\omega(S_M)$, $M_\omega(v)$, and $\rho_\omega^v(S_T)$, the same argument carries over and completes the proof. □



(a) Graph G (b) Cascade ω_1 (c) Cascade ω_2

Fig. 2. Example for the MCA problem

This lemma establishes the Shapley value equivalence well beyond the CIC model, since a wide range of well-established diffusion models fit the setting of $\mathcal{M}$. For example, the temporal competitive independent cascade (TCIC) model [41], a CIC variant that accounts for diffusion delays and allows users to be activated after a delay upon receiving a campaign, is one such model.

2.4 Problem Definition

Definition 2.3 indicates that the mitigation contribution for each seed is calculated based on a specific cascade ω, which, however, may be difficult to characterize precisely in practice due to the privacy protections in OSNs [18]. In contrast, the activation statuses of nodes is more readily observable; for instance, it can be inferred from the posts that users share. As such, in this paper, we focus on the node-level feedback setting, where edge statuses are unavailable and only node activations are provided. Specifically, we assume that an observation $\mathcal{T} = \{\Psi(S_M), \Psi(S_T), t\}$, obtained from a random CIC diffusion process, is given. Here, $\Psi(S_M)$ (resp. $\Psi(S_T)$) is the set of non-seed nodes activated by the campaign M (resp. T), and t is a function mapping each activated node to its activation timestamp. Note that there may exist multiple cascades whose node statuses are consistent with a given observation $\mathcal{T}$; we term each such cascade *valid* and denote it by $\omega \sim \mathcal{T}$. Accordingly, our goal is to compute the expected value of $\sigma_\omega(s)$ for each $s \in S_T$ over all valid cascades. The formal problem definition is provided below.

Problem statement. Given a directed graph G, a misinformation seed set S_M, a truth seed set S_T[1] with $S_T \cap S_M = \emptyset$, and an observation $\mathcal{T}$, the *mitigation contribution allocation* (MCA) problem aims to compute $\mathbb{E}_{\omega\sim\mathcal{T}}[\sigma_\omega(s)]$ for each $s \in S_T$, under the CIC model.

For brevity, we use $\mathbb{E}_{\mathcal{T}}[\sigma(s)]$ to denote $\mathbb{E}_{\omega\sim\mathcal{T}}[\sigma_\omega(s)]$ hereafter.

Example 2.7. To illustrate the MCA problem end-to-end, we consider the graph G in Figure 2(a) with seed sets $S_M = \{s_0\}$ and $S_T = \{s_1, s_2\}$, where each edge is labeled with its propagation probability. We set the tie-breaking probability $p_T^{v_0} = 1$ so that v_0 adopts T whenever both campaigns arrive, and suppose the observation $\mathcal{T}$ records v_0 as T-active. Since edges $\langle s_0, v_0\rangle$ and $\langle s_1, v_0\rangle$ both have probability probability 1, v_0 always receives M from s_0 and T from s_1, and is therefore T-active regardless of the status of $\langle s_2, v_0\rangle$. Consequently, both possible cascades of G, ω_1 and ω_2 shown in Figures 2(b) and (c), are valid, each with conditional probability $\Pr[\omega \mid \mathcal{T}] = 1/2$.

- In cascade ω_1, $\langle s_2, v_0\rangle$ is dead, so only s_1 delivers T to v_0; hence $M_{\omega_1}(v_0) = \{s_1\}$, giving $\sigma_{\omega_1}(s_1) = \Delta$ and $\sigma_{\omega_1}(s_2) = 0$.
- In cascade ω_2, $\langle s_2, v_0\rangle$ is live, so both s_1 and s_2 deliver T to v_0; hence $M_{\omega_2}(v_0) = \{s_1, s_2\}$, and the equal-share rule splits the credit evenly to give $\sigma_{\omega_2}(s_1) = \sigma_{\omega_2}(s_2) = \Delta/2$.

[1] S_T corresponds to the influencers who accept the invitation from OSN provider and execute the mitigation campaign. In practice, an invited influencer may attempt to extort the provider before accepting, e.g., by threatening to promote misinformation unless offered a higher reward; such behavior can be handled by platform-level sanctions, e.g., account suspension or reduced content visibility.

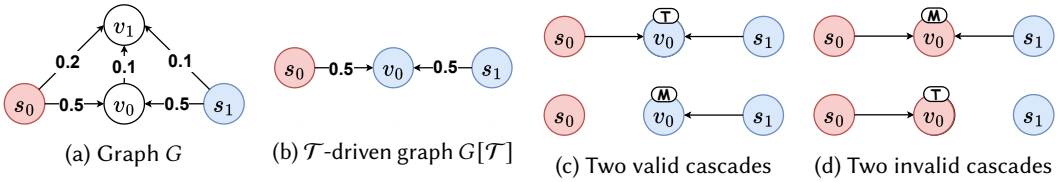


Fig. 3. Example of the possible world-induced cascades

Averaging over the two valid cascades yields the MCA solution: $\mathbb{E}_{\mathcal{T}}[\sigma(s_1)] = \frac{1}{2}\Delta + \frac{1}{2} \cdot \frac{\Delta}{2} = \frac{3\Delta}{4}$ and $\mathbb{E}_{\mathcal{T}}[\sigma(s_2)] = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot \frac{\Delta}{2} = \frac{\Delta}{4}$. This result aligns with a fair allocation principle: influencers who are more likely to deliver the truth contribute more to mitigation and thus deserve higher rewards.

Remark. While we frame MCA in the context of misinformation mitigation, the underlying framework naturally extends to broader competitive diffusion scenarios. For example, in competitive influencer marketing where a smartphone brand (e.g., Apple) hires influencers to promote its products against a competitor (e.g., Samsung), MCA can fairly quantify each influencer's contribution to the observed campaign result and allocate the incentives accordingly.

3 Validity-Guaranteed Sampling

Before presenting our approach to the studied MCA problem, we first introduce several definitions to facilitate exposition.

Definition 3.1 (Observation-Driven Graph). Given a graph $G = (V, E)$ and an observation $\mathcal{T}$, the observation-driven graph, denoted by $G[\mathcal{T}] = (V[\mathcal{T}], E[\mathcal{T}])$, is the subgraph of G induced by $\mathcal{T}$. Specifically, the node set $V[\mathcal{T}]$ consists of all activated nodes, and the edge set $E[\mathcal{T}]$ is expressed as $\{\langle u, v\rangle \in E \mid t(u) = t(v) - 1\}$.

Recall that $t(u) = t(v) - 1$ is the necessary condition for an edge $\langle u, v\rangle$ to be included in a cascade. It follows that $G[\mathcal{T}]$ characterizes all the potential CIC cascades for any given observation $\mathcal{T}$.

We then introduce the concept of possible world, which pre-determines all random events in a CIC diffusion process. For ease of presentation in the rest of this paper, we adopt a uniform setting for the tie-breaking probabilities, i.e., $p_M^v \equiv p_M$ and $p_T^v \equiv p_T$ for all $v \in V$. Note that this is purely a notational convenience; our proposed algorithms apply directly to the general node-specific case without compromising any theoretical guarantees.

Definition 3.2 (Possible World). Given an observation-driven graph $G[\mathcal{T}]$, its possible world, denoted by g, is a random subgraph of $G[\mathcal{T}]$ generated based on edge diffusion and node tie-breaking probabilities. Specifically, each edge $\langle u, v\rangle \in E[\mathcal{T}]$ is retained in the edge set E_g of g with probability $p(u, v)$; we refer to such retained edges as *live*. Besides, each node $v \in V[\mathcal{T}]$ is assigned a random campaign label r_v, drawn as M (resp. T) with probability p_M (resp. p_T). Formally, g is expressed as $(\mathcal{V}_g, E_g)$, where $\mathcal{V}_g = \{(v, r_v) \mid v \in V[\mathcal{T}]\}$.

The diffusion initiated from S_T and S_M under a possible world g yields a deterministic cascade, which we denote by $\omega(g)$. In particular, when a node v is reachable by both campaigns under $\omega(g)$, it adopts the campaign r_v. Conversely, if v is reachable by only one campaign, it directly adopts that campaign, independent of r_v.

Example 3.3. Consider the graph G in Figure 3(a), where each edge is labeled with its propagation probability. The seed sets are $S_M = \{s_0\}$ and $S_T = \{s_1\}$, and the tie-breaking probabilities are $p_M = p_T = 0.5$. Given an observation $\mathcal{T}$ where $\Psi(S_M) = \emptyset$, $\Psi(S_T) = \{v_0\}$, $t(s_0) = t(s_1) = 0$, and $t(v_0) = 1$, the corresponding observation-driven graph $G[\mathcal{T}]$ is shown in Figure 3(b). Figures 3(c)-(d)

further illustrate four cascades derived from possible worlds of $G[\mathcal{T}]$, where the label above v_0 denotes its campaign label r_{v_0}. Since $\mathcal{T}$ requires v_0 to be T-activated (shown in blue), the cascades in Figure 3(c) are valid, whereas those in Figure 3(d) are invalid.

Given an observation-driven graph $G[\mathcal{T}]$, let $m_{\mathcal{T}} = |E[\mathcal{T}]|$ and $n_{\mathcal{T}} = |V[\mathcal{T}]|$ denote the number of edges and nodes in $G[\mathcal{T}]$, respectively. It can be identified that there are exponential number $O(2^{m_{\mathcal{T}}+n_{\mathcal{T}}})$ of potential cascades. This implies that exhaustively enumerating all valid cascades from $G[\mathcal{T}]$ to exactly solve the MCA problem (i.e., compute the expected mitigation contributions) is prohibitively expensive. A more practical alternative is to approximate the contributions through sampling. However, enabling efficient and scalable approximation with accuracy guarantees is far from trivial, with a critical hurdle being the unbiased generation of valid cascades. Notably, a straightforward sampling method, which independently specifies the campaign label of each node and the status (live or not) of each edge, is nearly impossible to produce valid cascades. To explain intuitively, a cascade typically involves a substantial number of activated nodes, and it becomes invalid once the activation campaign of any one of these nodes deviates from the observation. As a result, the Reject-Sampling approach, which employs the straightforward sampling method to draw cascades from $G[\mathcal{T}]$ until a valid one is obtained, is practically infeasible.

In this section, we address this challenge by proposing Validity-Guaranteed (VG)-Sampling, an efficient algorithm that strictly generates valid cascades. Meanwhile, by maintaining lightweight auxiliary information during the sampling process, VG-Sampling enables the computation of the mitigation contribution for each seed under the generated cascade with negligible computational overhead. In Section 3.1, we first outline the motivation and the general idea of VG-Sampling. We then detail its algorithmic implementation in Section 3.2. Finally, we provide a comprehensive analysis of its correctness, effectiveness, and time complexity in Section 3.3.

3.1 Motivation and General Idea

Given an observation $\mathcal{T}$, for any activated node v, let $C_v^+ \in \{M, T\}$ denote v's activation campaign (i.e., M if $v \in \Psi(S_M)$ and T otherwise), and $C_v^- = \{T, M\} \setminus \{C_v^+\}$ denote the opposing campaign. Moreover, for a cascade $\omega(g)$ obtained from $G[\mathcal{T}]$ and an activated node v, we define the following two events to characterize the scenarios where v is activated by C_v^+ in $\omega(g)$:

- A_g^v: Node v receives both campaigns in $\omega(g)$, and the campaign label r_v of v is C_v^+ in g.
- B_g^v: Node v receives only the C_v^+ campaign in $\omega(g)$.

The sample generation procedure of VG-Sampling is grounded in the following lemma, which establishes a necessary and sufficient condition for producing a valid cascade $\omega(g) \sim \mathcal{T}$.

Lemma 3.4. *A cascade $\omega(g)$ sampled from $G[\mathcal{T}]$ is valid if and only if, for every activated non-seed node v, either A_g^v or B_g^v occurs.*

Proof sketch. We prove the two directions separately. ($\Rightarrow$) If $\omega(g)$ is valid, CIC semantics force each activated v to receive C_v^+, and whether it additionally receives C_v^- determines whether A_g^v or B_g^v occurs. ($\Leftarrow$) Given $A_g^v \vee B_g^v$ at every activated non-seed node, it suffices to show each activated v is reachable from $S_{C_v^+}$ under $\omega(g)$, which we establish by tracing back from v along C_v^+-activated edges in $\omega(g)$: since $G[\mathcal{T}]$ is a DAG, this trace eventually reaches a seed in $S_{C_v^+}$. The full proof is deferred to the online appendix [1]. □

Denote $E_{in}(v)$ by the incoming edges of node v in $G[\mathcal{T}]$. The key insight from Lemma 3.4 is that it reduces the intractable global sampling task to a collection of independent edge-selection problems: for each activated non-seed node v, it suffices to select incoming edges in $E_{in}(v)$ such that $A_g^v \vee B_g^v$ occurs. This factorization naturally motivates the node-wise design of VG-Sampling. To realize this,

we further examine the condition $A_g^v \vee B_g^v$, which treats the two campaigns asymmetrically: v must have at least one incoming neighbor activated by C_v^+, while incoming neighbors activated by C_v^- are optional. This asymmetry motivates a partition-based edge-selection strategy, in which $E_{in}(v)$ is split into $E_{in}^+(v)$ and $E_{in}^-(v)$ and each subset is sampled under its own scheme. Here, $E_{in}^+(v)$ (resp. $E_{in}^-(v)$) denotes a subset of $E_{in}(v)$ consisting of edges whose sources are activated by C_v^+ (resp. C_v^-).

Edge selection in $E_{in}^+(v)$. As discussed above, at least one edge from $E_{in}^+(v)$ should be selected and incorporated into the cascade. To this end, we first assume a fixed order on the C_v^+-activated incoming neighbors of v in $G[\mathcal{T}]$, written as $V_{in}^+(v) = \{u_1^+, u_2^+, \ldots, u_{l_v^+}^+\}$ where $l_v^+ = |V_{in}^+(v)|$. Let X_i^v be the event that $\langle u_i^+, v\rangle$ is the first edge in $E_{in}^+(v)$ to be selected, X^v be the event that at least one edge in $E_{in}^+(v)$ is selected, and β_v be the probability that X^v occurs. As the status of each edge is determined independently, it follows that

$$\Pr[X_i^v] = p(u_i^+, v) \cdot \prod_{j=1}^{i-1} \Big(1 - p(u_j^+, v)\Big),$$

$$\beta_v = \Pr[X^v] = \sum_{i=1}^{l_v^+} \Pr[X_i^v] = 1 - \prod_{i=1}^{l_v^+} \big(1 - p(u_i^+, v)\big).$$

On this basis, the first edge $\langle u_i^+, v\rangle$ in $E_{in}^+(v)$ can be selected according to the probability distribution of $\Pr[X_i^v]/\beta_v$. Then, the standard sampling procedure is applied to examine the remaining edges, i.e., each edge $\langle u_k^+, v\rangle$ is live with $p(u_k^+, v)$ probability for $i < k \le l_v^+$.

Edge selection in $E_{in}^-(v)$. In contrast to the case of $E_{in}^+(v)$, the selection from $E_{in}^-(v)$ allows for either selecting at least one edge or none at all. Assume a fixed order on the C_v^--activated incoming neighbors of v in $G[\mathcal{T}]$ as $V_{in}^-(v) = \{u_1^-, u_2^-, \ldots, u_{l_v^-}^-\}$ where $l_v^- = |V_{in}^-(v)|$. Let Y_i^v be the event that $\langle u_i^-, v\rangle$ is the first edge in $E_{in}^-(v)$ to be selected. Let Y^v be the event that at least one edge in $E_{in}^-(v)$ is selected, and let $\neg Y^v$ denote its complement, representing the event where no edge in $E_{in}^-(v)$ is selected, i.e., $\Pr[Y^v] = 1 - \Pr[\neg Y^v]$. Analogous to the probability expressions for X_i^v and X^v, we obtain

$$\Pr[Y_i^v] = p(u_i^-, v) \cdot \prod_{j=1}^{i-1} \Big(1 - p(u_j^-, v)\Big),$$

$$\Pr[Y^v] = 1 - \prod_{i=1}^{l_v^-} \big(1 - p(u_i^-, v)\big),\ \Pr[\neg Y^v] = \prod_{i=1}^{l_v^-} \big(1 - p(u_i^-, v)\big).$$

Besides, let Z^v be the event that the campaign label of v is C_v^+ in g. We have $\Pr[Z^v] = p_{C_v^+}$. Conditional on the occurrence of X^v, it can be verified that $Y^v \cap Z^v$ corresponds to A_g^v, and $\neg Y^v$ corresponds to B_g^v. Hence, for A_g^v or B_g^v to occur, the remaining task is to select edges from $E_{in}^-(v)$ such that either $Y^v \cap Z^v$ or $\neg Y^v$ happens. Since Y^v and Z^v are independent, we have $\Pr[Y^v \cap Z^v] = \Pr[Y^v] \cdot \Pr[Z^v]$. Furthermore, as $Y^v \cap Z^v$ and $\neg Y^v$ are disjoint, the probability γ_v that either occurs is given by

$$\begin{aligned}\gamma_v &= \Pr\big[(Y^v \cap Z^v) \cup \neg Y^v\big] = \Pr[Y^v \cap Z^v] + \Pr[\neg Y^v].\\ &= \Pr[Y^v] \cdot p_{C_v^+} + \Pr[\neg Y^v].\end{aligned}$$

Building on this, with probability $\Pr[\neg Y^v]/\gamma_v$, the selection process in $E_{in}^-(v)$ terminates immediately. With the other probability $\Pr[Y^v] \cdot p_{C_v^+}/\gamma_v$, the first edge $\langle u_i^-, v\rangle$ in $E_{in}^-(v)$ is selected according to the distribution $\Pr[Y_i^v] \cdot p_{C_v^+}/\gamma_v$, after which the standard sampling procedure is utilized to the remaining edges.

Algorithm 1: VG-Sampling

Input : The graph G, seed sets S_M, S_T and the observation $\mathcal{T}$.
Output: The value $\sigma_\omega(s)$ for each $s \in S_T$ under a valid ω.

for each $v \in \Psi(S_M) \cup \Psi(S_T)$ **do**
  $M_\omega(v) \leftarrow \emptyset$, $\rho^v_\omega(S_T) \leftarrow 0$, $Reach^\omega_M(v) \leftarrow 0$;
for each $s \in S_T$ **do**
  $M_\omega(s) \leftarrow \{s\}$, $\sigma_\omega(s) \leftarrow 0$;
for each $s \in S_M$ **do** $Reach^\omega_M(s) \leftarrow 1$;
$t_{\max} \leftarrow \max_{u \in \Psi(S_M) \cup \Psi(S_T)} t(u)$;
for $j \leftarrow 1$ *to* $t_{\max}$ **do**
  for each $v \in \Psi(S_M) \cup \Psi(S_T)$ *with* $t(v) = j$ **do**
    $I^+ \leftarrow$ EdgeSelection+$(G[\mathcal{T}], v, Reach^\omega_M)$;
    $I^- \leftarrow$ EdgeSelection-$(G[\mathcal{T}], v, Reach^\omega_M)$;
    if $v \in \Psi(S_T)$ **then**
      $\rho^v_\omega(S_T) \leftarrow \Delta$;
      for each $u^+_k \in V^+_{in}(v)$ *with* $k \in I^+$ **do**
        $M_\omega(v) \leftarrow M_\omega(v) \cup M_\omega(u^+_k)$;
    else if $v \in \Psi(S_M)$ *and* $I^- \neq \emptyset$ **then**
      $\rho^v_\omega(S_T) \leftarrow \Lambda$;
      for each $u^-_k \in V^-_{in}(v)$ *with* $k \in I^-$ **do**
        $M_\omega(v) \leftarrow M_\omega(v) \cup M_\omega(u^-_k)$;
for each $v \in \Psi(S_M) \cup \Psi(S_T)$ *with* $Reach^\omega_M(v) = 1$ **do**
  for each $s \in M_\omega(v)$ **do**
    $\sigma_\omega(s) \leftarrow \sigma_\omega(s) + \frac{\rho^v_\omega(S_T)}{|M_\omega(v)|}$;
return $\{\sigma_\omega(s) : s \in S_T\}$;

Efficient computation. To compute the mitigation contribution $\sigma_\omega(s)$ for each $s \in S_T$ under a valid $\omega \sim \mathcal{T}$, a direct method is to first conduct a BFS from S_T and S_M in ω to determine $R_\omega(S_M)$, i.e., the set of non-seed nodes activated by M in the absence of T, and $\Psi_\omega(s)$, i.e., the set of nodes receiving T from s. Meanwhile, compute the mitigation $\rho^v_\omega(S_T)$ achieved by S_T for each $v \in R_\omega(S_M) \cap \Psi_\omega(s)$. Then, for each such v, determine $M_\omega(v)$, i.e., the set of seeds in S_T that can enable v to receive T, by performing a reverse BFS from v restricted to T-activated incoming neighbors.

Evidently, this approach suffers from a major bottleneck caused by repeated reverse BFS traversals. To facilitate the calculation, our key idea is to determine the required node sets during the sampling procedure by processing nodes in ascending order of their activation timestamps. In this way, the sets associated with each node can be recursively constructed from the sets of its incoming neighbors.

3.2 Detailed Implementation

The pseudocode of VG-Sampling is outlined in Algorithm 1. It starts with an initialization phase. For each truth seed $s \in S_T$, its mitigation contribution $\sigma_\omega(s)$ is initialized to 0, and $M_\omega(s)$ is set to contain the seed s itself. Also, for each activated non-seed node v, $M_\omega(v)$ is initialized as an empty set, while $\rho^v_\omega(S_T)$ and $Reach^\omega_M(v)$ are both set to 0. Here, $Reach^\omega_M(v)$ is a flag indicating

Algorithm 2: EdgeSelection+$(G[\mathcal{T}], v, Reach_M^\omega)$

$I^+ \leftarrow$ an empty set for recording node indices;
$r_1 \leftarrow$ a random number selected uniformly from $[0, 1]$;
Find the lowest index i such that $\sum_{j=1}^{i} \frac{\Pr[X_j^v]}{\beta_v} \geq r_1$ via binary search and insert i into I^+;
for each $i + 1 \leq k \leq l_v^+$ **do**
    $r_2 \leftarrow$ a random number selected uniformly from $[0, 1]$;
    if $r_2 \leq p(u_k^+, v)$ **then** Insert k into I^+;
for each $u_k^+ \in V_{in}^+(v)$ *with* $k \in I^+$ **do**
    if $Reach_M^\omega(u_k^+) = 1$ **then** $Reach_M^\omega(v) = 1$, **break**;
return I^+;

whether v belongs to $R_\omega(S_M)$; accordingly, $Reach_M^\omega(s)$ is set to 1 for each misinformation seed $s \in S_M$ (Lines 1-5). Next, the non-seed activated nodes are examined by their activation timestamps in ascending order (Lines 7-8). For each activated non-seed node v, it first invokes EdgeSelection+ (Algorithm 2) and EdgeSelection- (Algorithm 3), to identify the live incoming edges of v in ω, following the probabilistic scheme described in Section 3.1 (Lines 9-10). Meanwhile, $Reach_M^\omega(v)$ is set to 1 if there exists at least one node u within the v's incoming neighbors in ω that satisfies $Reach_M^\omega(u) = 1$. Then, $\rho_\omega^v(S_T)$ is computed according to Eq. (1), and $M_\omega(v)$ is determined by inheriting the corresponding sets from v's T-activated incoming neighbors in ω (Lines 11-18). Once all nodes have been processed, for each activated non-seed v with $Reach_M^\omega(v) = 1$, the algorithm iterates through $M_\omega(v)$, distributing an equal share of $\rho_\omega^v(S_T)$ to each $s \in M_\omega(v)$ by adding $\frac{\rho_\omega^v(S_T)}{|M_\omega(v)|}$ to $\sigma_\omega(s)$ (Lines 19-21). Finally, the mitigation contributions of all truth seeds under a valid cascade, i.e., $\{\sigma_\omega(s) : s \in S_T\}$, are returned (Line 22).

3.3 Theoretical Analysis

Correctness. Let $\Pr[\omega \sim \mathcal{T}]$ be the probability of sampling a random valid cascade ω using our VG-Sampling method, and let $\Pr[\omega \sim G[\mathcal{T}]]$ be the probability of sampling ω from the original sample space, i.e., by employing the straightforward sampling approach on $G[\mathcal{T}]$ to independently determine the statuses of each node and edge. The following lemma establishes the unbiasedness of the cascade generation in VG-Sampling.

Lemma 3.5. *For a given observation $\mathcal{T}$, the random cascades generated by the VG-Sampling approach follow probabilities proportional to those in the original sample space of cascades. That is, for any two valid cascades ω_1 and ω_2 under $\mathcal{T}$, we have* $\frac{\Pr[\omega_1 \sim \mathcal{T}]}{\Pr[\omega_2 \sim \mathcal{T}]} = \frac{\Pr[\omega_1 \sim G[\mathcal{T}]]}{\Pr[\omega_2 \sim G[\mathcal{T}]]}$.

Proof Sketch. By the independence of edge-selection across nodes under both VG-Sampling and straightforward sampling, the probability ratio between ω_1 and ω_2 factors into a product of per-node ratios. It thus suffices to show that this ratio matches under the two sampling schemes at every activated non-seed node v. We verify this via a case analysis on whether v has any live incoming edge from the opposing campaign in ω_1 and ω_2, which determines the exact form of the per-node probabilities under VG-Sampling. The full proof is deferred to the online appendix [1]. □

Effectiveness. For any valid cascade ω, the lemma below then presents the relationship between $\Pr[\omega \sim \mathcal{T}]$ and $\Pr[\omega \sim G[\mathcal{T}]]$.

Algorithm 3: EdgeSelection-$(G[\mathcal{T}], v, Reach_M^{\omega})$

$I^- \leftarrow$ an empty set for recording node indices;
$r_1 \leftarrow$ a random number selected uniformly from $[0, 1]$;
if $r_1 > \frac{\Pr[Y^v] \cdot p_{C_v^+}}{\gamma_v}$ **then return** I^-;
Find the lowest index i such that $\sum_{j=1}^{i} \frac{\Pr[Y_j^v] \cdot p_{C_v^+}}{\gamma_v} \geq r_1$ via binary search and insert i into I^-;
for each $i + 1 \leq k \leq l_v^-$ **do**
    $r_2 \leftarrow$ a random number selected uniformly from $[0, 1]$;
    if $r_2 \leq p(u_k^-, v)$ **then** Insert k into I^-;
if $Reach_M^{\omega}(v) = 0$ **then**
    for each $u_k^- \in V_{in}^-(v)$ *with* $k \in I^-$ **do**
        if $Reach_M^{\omega}(u_k^-) = 1$ **then** $Reach_M^{\omega}(v) = 1$, **break**;
return I^-;

Lemma 3.6. *For each valid cascade ω under the given $\mathcal{T}$, we have*

$$\Pr[\omega \sim \mathcal{T}] = \frac{1}{\prod_{v \in \Psi(S_M) \cup \Psi(S_T)} \beta_v \gamma_v} \cdot \Pr[\omega \sim G[\mathcal{T}]].$$

Proof Sketch. By Lemma 3.4, a cascade is valid if and only if each node v satisfies $A_g^v \vee B_g^v$, which has probability $\beta_v \gamma_v$ under straightforward sampling. Thus the total probability of valid cascades in the original space is $\prod_v \beta_v \gamma_v$. Combined with Lemma 3.5 and the fact that VG-Sampling produces only valid cascades, the lemma follows. The full proof is deferred to online appendix [1]. □

This result highlights the effectiveness of our VG-Sampling approach in terms of valid cascade sampling. In particular, it characterizes the extent to which VG-Sampling shrinks the original sample space, achieving a compression ratio of $\frac{1}{\prod_{v \in \Psi(S_M) \cup \Psi(S_T)} \beta_v \gamma_v}$. This quantity often reaches astronomical magnitudes in practice, as we reported in Table 4 of Section 5.

Time complexity. Recall that $m_{\mathcal{T}}$ and $n_{\mathcal{T}}$ are the number of edges and nodes in $G[\mathcal{T}]$, respectively. The main computational overhead of VG-Sampling lies in three tasks: (i) generating cascade ω; (ii) updating $M_\omega(v)$ for each activated non-seed v; and (iii) computing $\sigma_\omega(s)$ for each $s \in S_T$. It is trivial to obtain that the first task requires $O(m_{\mathcal{T}})$ runtime, and the third task takes $O(n_{\mathcal{T}} \cdot |S_T|)$ runtime. Regarding the second task, $M_\omega(v)$ will be updated once a new live incoming edge with the T-activated source is selected. Each such update corresponds to a union operation, which incurs $O(|S_T|)$ time. In addition, the total number of live edges in a cascade is $O(m_{\mathcal{T}})$ in the worst case. Hence, the second task requires $O(m_{\mathcal{T}} \cdot |S_T|)$ runtime. To sum up, the time complexity of VG-Sampling is $O(m_{\mathcal{T}} \cdot |S_T|)$.

4 Approximation with Provable Guarantees

This section designs VG-Sampling-based Approximation (VG-App), an efficient algorithm for the MCA problem that approximates the mitigation contributions of all truth seeds with provable accuracy guarantees. VG-App strikes a balance between accuracy and efficiency, by progressively generating cascades (via VG-Sampling) until all seed-specific stopping conditions are met. This sampling scheme follows the stopping rule framework [69], a general and well-established technique for estimation tasks, yet incorporates new stopping criteria tailored to the mitigation contribution function. In Section 4.1, we first briefly revisit the stopping rule framework and then elaborate on

Algorithm 4: VG-App

Input : The graph G, seed sets S_M, S_T, the observation $\mathcal{T}$, and the error parameters ϵ, δ.
Output: The (ϵ, δ)-estimate $\hat{\sigma}(s)$ of the expected mitigation contribution $\mathbb{E}_{\mathcal{T}}[\sigma(s)]$ for each $s \in S_T$.

1 $\theta \leftarrow 0$;
2 $\{b_s : s \in S_T\} \leftarrow$ BoundComputation$(G[\mathcal{T}])$;
3 **for each** $s \in S_T$ **do**
4 $\Sigma_s \leftarrow 0,\ \eta_s \leftarrow \mathbb{T}(0, b_s, \epsilon, \delta)$;
5 **while** $\exists s \in S_T$ *such that* $\Sigma_s < \eta_s$ **do**
6 $\{\sigma_\omega(s) : s \in S_T\} \leftarrow$ VG-Sampling$(G, S_M, S_T, \mathcal{T})$;
7 $\theta \leftarrow \theta + 1$;
8 **for each** $s \in S_T$ **do**
9 **if** $\Sigma_s < \eta_s$ **then**
10 $\Sigma_s \leftarrow \Sigma_s + \sigma_\omega(s),\ \theta_s \leftarrow \theta$;
11 **return** $\{\hat{\sigma}(s) = \frac{\eta_s}{\theta_s} : s \in S_T\}$;

the algorithmic design of VG-App. In Section 4.2, we further analyze the approximation guarantees and expected time complexity of VG-App.

4.1 The VG-App Algorithm

Stopping rule framework. Consider a random variable X bounded in the range $[a, b]$ with expectation $\mathbb{E}[X] = \mu_X$. Given two user-specified error parameters $\epsilon, \delta \in (0, 1)$, the stopping rule framework is able to yield an (ϵ, δ)-estimate $\hat{\mu}_X$ that satisfies the following probabilistic accuracy guarantee:

$$\Pr\left[(1-\epsilon)\mu_X \le \hat{\mu}_X \le (1+\epsilon)\mu_X\right] \ge 1 - \delta.$$

Specifically, instead of using a fixed sample size, the framework adopts a sequential sampling strategy. It iteratively generates a sequence of samples $X_1, X_2, \ldots$ and maintains a cumulative sum $\sum_{i=1}^{k} X_i$ at each step k. The process terminates once this sum exceeds a threshold $\mathbb{T}(a, b, \epsilon, \delta)$, which is formulated in terms of the variable's bounds a, b and the input error parameters ϵ, δ:

$$\mathbb{T}(a, b, \epsilon, \delta) = \frac{2}{\epsilon^2}(b-a)(1+\epsilon)\left(\frac{b-a}{b} + \frac{\epsilon}{3}\right)\ln\frac{2}{\delta}. \tag{4}$$

As established by Zhu et al. [69], the value $\hat{\mu}_X = \mathbb{T}(a, b, \epsilon, \delta)/\theta$ constitutes a (ϵ, δ)-estimate of μ_X, where θ denotes the total number of generated samples.

Detailed implementation. Instantiating the stopping rule framework to the MCA problem raises two questions: (i) how to obtain, for each seed, a tight upper bound b_s on $\sigma_\omega(s)$ across all possible valid cascades, and (ii) how to adapt the stopping condition to each seed, given that seeds may have different contributions. VG-App addresses the first question by a dedicated BoundComputation procedure, and the second by maintaining a per-seed cumulative counter and terminating independently for each seed.

Algorithm 4 presents the pseudocode of VG-App. It begins by invoking the BoundComputation procedure to compute b_s for each seed s (Line 2). Based on these bounds and the input accuracy parameters ϵ, δ, the threshold η_s for each $s \in S_T$ is calculated according to Eq. (4) (Lines 3-4). Afterward, the algorithm proceeds in an iterative manner. In each iteration, the VG-Sampling procedure is invoked to generate a new valid cascade and to compute the mitigation contribution

Algorithm 5: BoundComputation($G[\mathcal{T}]$)

for each $v \in \Psi(S_M) \cup \Psi(S_T)$ **do**
  $M(v) \leftarrow \emptyset$, $Reach_M(v) \leftarrow 0$;
for each $s \in S_T$ **do**
  $M(s) \leftarrow \{s\}$, $b_s \leftarrow 0$;
for each $s \in S_M$ **do** $Reach_M(s) \leftarrow 1$;
$t_{\max} \leftarrow \max_{u \in \Psi(S_M) \cup \Psi(S_T)} t(u)$;
for $j \leftarrow 1$ *to* $t_{\max}$ **do**
  for each $v \in \Psi(S_M) \cup \Psi(S_T)$ *with* $t(v) = j$ **do**
    // $V_{in}(v)$ denotes the incoming neighbors of v in $G[\mathcal{T}]$
    for each $u \in V_{in}(v) \cap \Psi(S_T)$ **do**
      $M(v) \leftarrow M(v) \cup M(u)$;
    for each $u \in V_{in}(v)$ **do**
      if $Reach_M(u) = 1$ **then** $Reach_M(v) = 1$, **break**;
for each $v \in \Psi(S_M) \cup \Psi(S_T)$ *with* $Reach_M(v) = 1$ **do**
  if $v \in \Psi(S_T)$ **then**
    for each $s \in M(v)$ **do**
      $b_s \leftarrow b_s + \Delta$;
  else if $v \in \Psi(S_M)$ **then**
    for each $s \in M(v)$ **do**
      $b_s \leftarrow b_s + \Lambda$;
return $\{b_s : s \in S_T\}$;

$\sigma_\omega(s)$ for each seed $s \in S_T$ (Line 6). The counter θ is then incremented to reflect the total number of cascades generated so far (Line 7). For every seed whose cumulative value Σ_s has not yet reached the threshold η_s, the algorithm updates Σ_s by adding the newly obtained contribution $\sigma_\omega(s)$ and records the corresponding sample size θ_s (Lines 8-10). The process terminates once every seed $s \in S_T$ satisfies $\Sigma_s \geq \eta_s$ (Line 5). At last, the estimation $\hat{\sigma}(s) = \eta_s/\theta_s$ is returned for each $s \in S_T$ (Line 11).

The details of BoundComputation are presented in Algorithm 5. At first, two essential structures are introduced and initialized: $M(v)$, which stores the seeds that can enable v to receive the truth in all possible valid cascades; and $Reach_M(v)$, a flag indicating whether v is reachable from S_M in $G[\mathcal{T}]$ when $S_T = \emptyset$ (Lines 1-5). The algorithm next iteratively updates both structures by processing nodes in ascending order of their activation timestamps (Lines 7-12). Specifically, for each node v, the set $M(v)$ is updated by aggregating the sets $M(u)$ from every T-activated incoming neighbor u in $G[\mathcal{T}]$. Also, $Reach_M(v)$ is set to 1 if any incoming neighbor of v is reachable from S_M in $G[\mathcal{T}]$. Subsequently, it traverses each activated non-seed node v with $Reach_M(v) = 1$ to calculate the upper bound b_s for every seed. In particular, for all $s \in M(v)$, b_s is increased by Δ if v is T-activated, and by Λ if v is M-activated (Lines 13-19). After the computation, the set of upper bounds $\{b_s : s \in S_T\}$ for each truth seed is then returned (Line 20).

4.2 Theoretical Analysis

Approximation guarantee. The following theorem presents the accuracy guarantee of the estimations returned by VG-App.

Table 2. Statistics of datasets

Dataset	Type	$\|V\|$	$\|E\|$	Avg. deg
Lastfm	directed	1,372	14,707	21.4
Facebook	undirected	4,039	88,234	43.7
Flixster	directed	28,843	119,018	9.6
Twitter	directed	81,306	1,768,149	43.5
LiveJournal	directed	4,847,571	68,993,773	28.5
Orkut	undirected	3,072,441	117,185,083	76.3

Theorem 4.1. *Given any error parameters $\epsilon, \delta \in (0,1)$, for each seed $s \in S_T$, the VG-App algorithm returns an (ϵ,δ)-estimate $\hat{\sigma}(s)$ of $\mathbb{E}_{\mathcal{T}}[\sigma(s)]$ such that*

$$\Pr\left[(1-\epsilon)\mathbb{E}_{\mathcal{T}}[\sigma(s)] \le \hat{\sigma}(s) \le (1+\epsilon)\mathbb{E}_{\mathcal{T}}[\sigma(s)]\right] \ge 1-\delta.$$

Proof Sketch. By the results of stopping rule framework stated earlier, it suffices to show that $\sigma_\omega(s) \in [0, b_s]$ for every valid cascade ω. Non-negativity is immediate; the upper bound follows because b_s aggregates the maximum possible credit s could earn at each node across all cascades, which therefore dominates $\sigma_\omega(s)$ for any specific ω. The full proof is deferred to the online appendix [1]. □

Time complexity. The main computational overhead of the VG-App algorithm arises from two tasks: (i) the execution of BoundComputation, and (ii) multiple invocations of VG-Sampling. The first task requires $O(m_{\mathcal{T}} \cdot |S_T|)$ time, as in the worst case, a union operation is performed for every incoming neighbor of each activated node. For the second task, it can be verified that the expected number of invocations is $O(\max_{s\in S_T} \frac{\eta_s}{\mathbb{E}_{\mathcal{T}}[\sigma(s)]})$. Furthermore, as established in Section 3.3, the time complexity of VG-Sampling is $O(m_{\mathcal{T}} \cdot |S_T|)$. In summary, we can derive that the expected time complexity of VG-App is

$$O(\max_{s\in S_T} \frac{\eta_s}{\mathbb{E}_{\mathcal{T}}[\sigma(s)]} m_{\mathcal{T}}|S_T|) = O(\max_{s\in S_T} \frac{\ln 1/\delta \cdot b_s/\epsilon^2}{\mathbb{E}_{\mathcal{T}}[\sigma(s)]} m_{\mathcal{T}}|S_T|)$$
$$= O(\max_{s\in S_T} \frac{\ln 1/\delta \cdot (|\Psi(S_T)|\Delta + |\Psi(S_M)|\Lambda)}{\mathbb{E}_{\mathcal{T}}[\sigma(s)] \cdot \epsilon^2} m_{\mathcal{T}}|S_T|).$$

Remark. Although designed for MCA, our techniques readily extend to estimating other conditional quantities under a given observation $\mathcal{T}$, with the same provable guarantees. For instance, by using VG-Sampling to draw cascades and scoring each cascade by the number of nodes reachable from the seed, we can quantify the influence of each seed conditioned on $\mathcal{T}$ in competitive diffusion.

5 Experiments

This section conducts extensive experiments to validate the performance of our proposed techniques. 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].

5.1 Experimental Settings

Algorithms. To evaluate the performance of VG-App, we implement the following three cascade-sampling-based baselines.

- UNI: The contributions are uniformly assigned to the seeds.
- DEG: The contributions are assigned to the seeds according to their degree distribution in G.

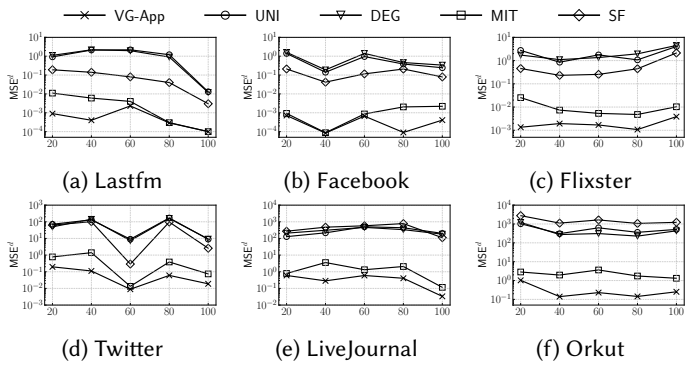


Fig. 4. MSE^d against the default ground truth under varying $|S_M|$

- MIT: The contributions are assigned to the seeds according to the distribution of their achieved mitigation.

It is worth noting that whether a tie-breaking event occurred at each misinformation-active node is not revealed by the node-level observation $\mathcal{T}$, yet it directly affects $\rho_\omega^v(S_T)$ via Eq. (1). As a consequence, the ideal normalizer $\rho_\omega(S_T)$ (i.e., the overall mitigation) for the three baselines cannot be determined from $\mathcal{T}$ alone. To address this hurdle, we compute $\hat{\rho}(S_T)$ by averaging $\rho_\omega(S_T)$ over 10,000 valid cascades sampled under $\mathcal{T}$, and normalize each baseline's total contributions to $\hat{\rho}(S_T)$. For MIT, which depends on per-seed mitigation that is also unavailable from $\mathcal{T}$, we additionally compute $\hat{\rho}(s)$ for each seed $s \in S_T$ by averaging $\rho_\omega(s)$ over the same 10,000 cascades, and allocate to each seed s a fraction $\hat{\rho}(s)/\sum_{s' \in S_T} \hat{\rho}(s')$.

We also implement a sampling-free baseline, SF, which analytically estimates contributions from the observation $\mathcal{T}$. For each non-seed activated node v, SF first assigns a credit $c(v)$ depending on v's activation status:

If $v \in \Psi(S_T)$, the Δ-mitigation at v is realized only if v is M-reached in the absence of S_T. We thus set $c(v) = \Delta \cdot q(v)$, where $q(v)$ approximates this probability, computed in topological order on the observation-driven graph $G[\mathcal{T}]$ via

$$q(v) = 1 - \prod_{u \in V_{\mathrm{in}}(v)} \big(1 - p(u,v) \cdot q(u)\big),$$

with $V_{\mathrm{in}}(v)$ denoting the in-neighbors of v in $G[\mathcal{T}]$ and $q(u) = 1$ for every M-activated node u.

If $v \in \Psi(S_M)$, the Λ-mitigation at v is realized only when a tie-break event occurs. Accordingly, we set $c(v)$ by weighting Λ with the conditional probability of such an event under the observation $\mathcal{T}$:

$$c(v) = \Lambda \cdot \frac{p_M \cdot \Pr[Y_v]}{p_M \cdot \Pr[Y_v] + (1 - \Pr[Y_v])},$$

where $\Pr[Y_v]$ denotes the probability that v is reached by at least one T-activated incoming neighbor on $G[\mathcal{T}]$, and p_M is the probability that the tie-break favors M.

Then, for each v, SF identifies the set $M_{\mathcal{T}}(v) \subseteq S_T$ of truth seeds that can reach v in $G[\mathcal{T}]$ via T-activated intermediates, and splits $c(v)$ equally among them. Formally, the contribution of each truth seed s under SF is given by $\sum_{v:\, s \in M_{\mathcal{T}}(v)} \frac{c(v)}{|M_{\mathcal{T}}(v)|}$.

Datasets. We use 6 real-world datasets, with their details shown in Table 2. LastFM [7] is a social network in which users declare their musical interests and form friendships, while Flixster [5] is a social movie-rating network. Since both datasets provide action logs recording users' activities, we follow the method in [8] to learn the diffusion probabilities on edges from these logs. For the

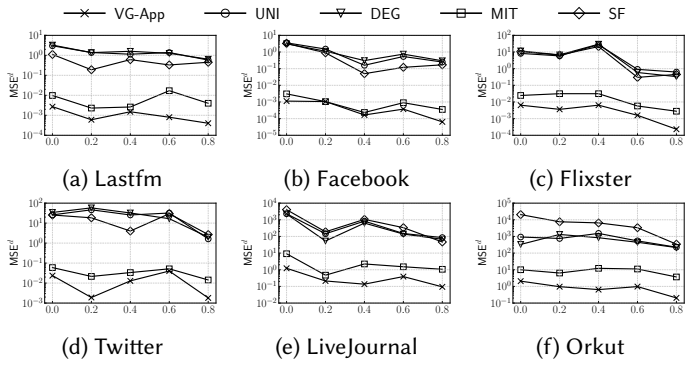


Fig. 5. MSE^d against the default ground truth under varying p_M

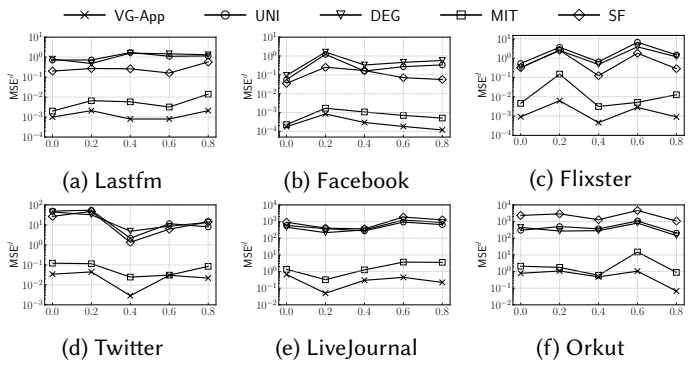


Fig. 6. MSE^d against the default ground truth under varying Λ

other datasets, including Facebook [32], Twitter [32], LiveJournal [6], and Orkut [67], which lack empirical action logs, we follow the convention [23, 28, 48, 49, 51, 55, 59, 60] and set the diffusion probability $p(u, v)$ of each edge $\langle u, v\rangle$ to the inverse of v's in-degree.

Default parameters. We set the sizes of the misinformation and truth seed sets to 20 and 10, respectively, i.e., $|S_M| = 20$, $|S_T| = 10$. We construct S_M by selecting the nodes with the highest out-degrees in G, a setting under which S_M typically exhibits a large influence spread. Subsequently, given the determined S_M, we apply the existing misinformation mitigation method [41] to identify the corresponding truth seed set S_T. We perform a stochastic CIC diffusion process from the sets S_M and S_T to generate an observation $\mathcal{T}$. In addition, the tie-breaking probabilities p_M and p_T are set to 0.5. The parameters Δ and Λ in the mitigation metric are set to 1 and 0.5, respectively. For the approximation algorithm VG-App, we set the error parameters as $\epsilon = 0.2, \delta = 0.1$. For each parameter setting, we repeat each algorithm 10 times and report the average results. We terminate any algorithm that fails to complete within 24 hours and record its runtime as **INF**.

Ground truth. As the MCA problem targets the computation of $\mathbb{E}_{\mathcal{T}}[\sigma(s)]$ for each seed $s \in S_T$, we take its $(0.1, 0.01)$-estimate $\tilde{\sigma}(s)$ as the default ground truth, which serves as a tight approximation.

To strengthen construct validity, we additionally introduce a proxy ground truth $\sigma^*(s)$ based on users' reposting behavior. The intuition is that, in practice, a user simultaneously exposed to multiple truth posts typically reshares only one, often chosen at random, and this reshared post can be regarded as the one that shaped the user's stance. In particular, we use VG-Sampling

Table 3. MSE^d against default ground truth under varying ϵ

Algorithm	Lastfm	Facebook	Flixster	Twitter	LiveJournal	Orkut
UNI	0.920	1.402	2.740	71.02	128.2	1028
DEG	1.114	1.603	1.722	49.66	207.7	1221
MIT	0.011	0.021	0.025	0.778	0.788	2.841
SF	0.190	0.210	0.456	63.34	271.8	2763
VG-App, $\epsilon = 0.1$	0.001	0.001	0.001	0.033	0.176	0.504
VG-App, $\epsilon = 0.2$	0.001	0.012	0.003	0.048	0.598	1.025
VG-App, $\epsilon = 0.3$	0.001	0.024	0.004	0.121	0.893	1.577
VG-App, $\epsilon = 0.4$	0.004	0.033	0.005	0.132	1.079	1.782
VG-App, $\epsilon = 0.5$	0.003	0.054	0.008	0.335	1.649	3.894

to generate 100,000 valid cascades under $\mathcal{T}$. In each cascade, we simulate users' reposting via a label-propagation procedure, where each node carries a label indicating which truth seed shaped its stance. Initially, each seed $s \in S_T$ takes itself as its label. The labels then propagate forward along the live edges: when a non-seed node v is reached by truth-active incoming neighbors, v uniformly picks one of their labels at random and inherits it as its own. Upon termination of label propagation, for each non-seed user v that benefits from mitigation, v's credit $\rho_\omega^v(S_T)$ is attributed entirely to v's label. Averaging the resulting per-seed totals over all valid cascades yields the proxy contribution $\sigma^*(s)$ for each $s \in S_T$.

5.2 Experimental Results

Accuracy. Let $\{\hat{\sigma}(s)\}_{s\in S_T}$ be the results returned by the evaluated algorithm. We assess its accuracy by computing the mean-squared error (MSE) of $\hat{\sigma}$ against the default ground truth $\tilde{\sigma}$ and the proxy ground truth σ^*, denoted by MSE^d and MSE^p respectively:

$$\text{MSE}^d = \frac{1}{|S_T|}\sum_{s\in S_T}(\hat{\sigma}(s) - \tilde{\sigma}(s))^2, \qquad \text{MSE}^p = \frac{1}{|S_T|}\sum_{s\in S_T}(\hat{\sigma}(s) - \sigma^*(s))^2.$$

Figures 4, 5, and 6 report the MSE^d of VG-App and four heuristic baselines as $|S_M|$, p_M, and Λ vary, respectively. We make the following observations. First, VG-App consistently achieves the lowest MSE^d across all datasets and parameter settings, because it is specifically designed to approximate $\mathbb{E}_{\mathcal{T}}[\sigma(s)]$ with provable guarantees. Second, UNI and DEG often perform poorly, since they distribute contributions based on static properties and use the observed campaign result $\mathcal{T}$ only for total normalization. Third, SF exhibits two distinct behaviors depending on graph scale: on the three smaller datasets (Lastfm, Facebook, and Flixster), it outperforms UNI and DEG, since it derives contributions from $\mathcal{T}$ analytically rather than using $\mathcal{T}$ only as a normalizer; on the larger Twitter, LiveJournal, and Orkut, however, it degrades to a level comparable with or even worse than UNI and DEG. This is because SF approximates each $q(v)$ by recursively multiplying per-edge probabilities under an independence assumption, whose error accumulates along long activation paths and is amplified on denser, larger networks where such paths heavily overlap. Fourth, MIT attains the lowest MSE^d among all baselines, yet remains inferior to VG-App. The reason is that MIT allocates contributions by per-seed mitigation, which is conceptually closest to our formulation; however, the two methods differ in how they handle joint mitigation. When multiple seeds simultaneously mitigate the same node, VG-App splits the contribution evenly among them, whereas MIT credits each seed with the full mitigation value, thereby over-counting joint contributions.

Table 3 then reports the MSE^d values under varying the error parameter ϵ. For the heuristics whose performance is not affected by ϵ, we report their results only under the default settings. The MSE^d of VG-App increases as ϵ increases, because a larger ϵ yields a smaller threshold η_s (see Eq. (4)) for each seed in VG-App, which in turn reduces the number of generated samples and

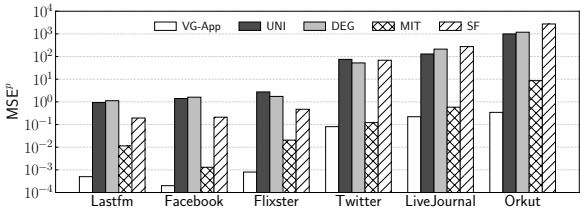


Fig. 7. MSE^p against the proxy ground truth

Table 4. $\mathcal{M}_1$ values under varying $|S_M|$

$\lvert S_M\rvert$	Lastfm	Facebook	Flixster	Twitter	LiveJournal	Orkut
20	$10^{75.8}$	$10^{334.7}$	$10^{106.9}$	$10^{574.9}$	$10^{3843.8}$	$10^{24713.9}$
40	$10^{59.6}$	$10^{174.4}$	$10^{123.8}$	$10^{584.8}$	$10^{2833.2}$	$10^{21297.1}$
60	$10^{38.2}$	$10^{313.3}$	$10^{160.2}$	$10^{679.8}$	$10^{1925.3}$	$10^{25023.7}$
80	$10^{25.9}$	$10^{246.9}$	$10^{126.5}$	$10^{673.5}$	$10^{2642.5}$	$10^{22656.2}$
100	$10^{14.9}$	$10^{300.0}$	$10^{131.9}$	$10^{561.1}$	$10^{1790.0}$	$10^{22656.7}$

lowers estimation accuracy, thereby increasing the MSE^d. In addition, VG-App achieves a lower MSE^d than MIT when $\epsilon \leq 0.2$, while for $\epsilon > 0.2$, MIT can yield a lower MSE^d on some datasets, e.g., on LiveJournal at $\epsilon = 0.5$, the MSE^d of VG-App is more than twice that of MIT. This indicates that the bias of MIT's allocation strategy becomes less harmful than VG-App's reduced sample size under large ϵ. Therefore, unless computational time becomes prohibitive, setting $\epsilon \leq 0.2$ is advisable to ensure reliable accuracy.

Figure 7 reports the MSE^p values against the proxy ground truth of all algorithms under the default parameter settings. As observed, the overall ordering aligns with that under MSE^d, with VG-App consistently achieving the lowest MSE^p. This superior performance arises from VG-App's principled handling of joint mitigation, the key advantage that distinguishes it from all baselines.

Techniques effectiveness. We proceed to empirically evaluate the effectiveness of the techniques incorporated in VG-App through two metrics. The first metric,

$$\mathcal{M}_1 = \frac{1}{\prod_{v \in \Psi(S_M) \cup \Psi(S_T)} \beta_v \gamma_v},$$

reflects the effectiveness of the VG-Sampling approach in terms of valid cascade sampling. In particular, according to Lemma 3.6, $\mathcal{M}_1$ corresponds to the factor by which VG-Sampling shrinks the original sample space. Equivalently, $\mathcal{M}_1$ represents the expected number of trials required for Reject-Sampling to obtain a valid cascade. Table 4 reports the values of $\mathcal{M}_1$ under varying the size $|S_M|$ of misinformation seed set. As we can see, $\mathcal{M}_1$ reaches astronomical magnitudes across all cases. Even on the smallest Lastfm dataset, the value still exceeds 10^{10}. On the largest Orkut, the factor grows to prohibitive levels, ranging from $10^{21,297}$ to $10^{25,023}$. These results underscore the practical infeasibility of Reject-Sampling, while highlighting the critical necessity and superior efficiency of our VG-Sampling method. Furthermore, we observe that $\mathcal{M}_1$ scales significantly with the graph size. This is attributed to the fact that larger networks typically involve a vastly greater number of activated nodes in $\Psi(S_M) \cup \Psi(S_T)$.

Recall that in VG-App, the threshold η_s for each seed s is defined as $\mathbb{T}(0, b_s, \epsilon, \delta)$, where b_s is derived by the BoundComputation procedure, and denotes the upper bound of s's mitigation contribution across all possible valid cascades. To accelerate the execution of VG-App, tighter upper bounds are desirable, as they result in lower thresholds and consequently fewer generated samples. To quantify the effectiveness of our BoundComputation procedure in tightening the bounds and reducing the sample size, we introduce $\mathcal{M}_2$, defined as the ratio between the threshold determined

Table 5. $\mathcal{M}_2$ values under varying $|S_M|$

$\lvert S_M\rvert$	Lastfm	**Facebook**	**Flixster**	**Twitter**	**LiveJournal**	**Orkut**
20	4.91	13.6	5.62	12.1	18.1	6.78
40	8.10	35.6	13.5	18.6	30.9	17.8
60	8.23	17.2	14.2	50.7	38.1	18.4
80	13.4	10.8	19.6	17.2	29.4	25.8
100	40.2	19.8	25.5	25.3	72.8	35.5

from a naive upper bound b_n and that from our derived upper bounds, i.e.,

$$\mathcal{M}_2 = \frac{\mathbb{T}(0, b_n, \epsilon, \delta)}{\max_{s \in S_T} \eta_s}.$$

Here, $b_n = |\Psi(S_T)|\Delta + |\Psi(S_M)|\Lambda$, which is directly obtained from the mitigation formulation in Eq. (1), where the mitigation for a T-activated (resp. M-activated) node is upper bounded by Δ (resp. Λ). We report $\mathcal{M}_2$ values under varying $|S_M|$ in Table 5. The results show that $\mathcal{M}_2$ often reaches into the tens, peaking at 72.8 on LiveJournal when $|S_M| = 100$. As such, BoundComputation plays a critical role in optimizing the overall efficiency of VG-App.

Table 6. Runtime (sec) under varying ϵ

Algorithm	**Lastfm**	**Facebook**	**Flixster**	**Twitter**	**LiveJournal**	**Orkut**
UNI	0.229	1.1	1.5	16.2	552.5	939.5
DEG	0.228	1.0	1.4	16.9	546.1	921.2
MIT	0.230	1.0	1.5	16.0	576.8	953.9
SF	0.0001	0.001	0.001	0.002	0.118	0.136
VG-App, $\epsilon = 0.1$	0.324	3.6	4.4	50.1	3790.8	24442.6
VG-App, $\epsilon = 0.2$	0.092	1.1	1.2	14.1	1059.4	6563.8
VG-App, $\epsilon = 0.3$	0.046	0.5	0.6	6.5	547.8	3277.5
VG-App, $\epsilon = 0.4$	0.028	0.3	0.4	4.1	346.2	2338.6
VG-App, $\epsilon = 0.5$	0.020	0.2	0.3	3.2	232.8	1511.7

Efficiency. We then report the running time of each algorithm under varying ϵ. As shown in Table 6, VG-App readily scales to all six datasets, and its runtime decreases as ϵ increases because a larger ϵ relaxes the stopping condition and reduces the required sample size. Notably, under loose ϵ, the heuristics UNI, DEG, and MIT can even be slower than VG-App, since they always draw a fixed 10,000 cascades, whereas VG-App may terminate much earlier. In contrast to all sampling-based methods, SF is orders of magnitude faster. For instance, on Orkut, SF finishes in 0.136 seconds, while every other method takes at least 900 seconds. This is because its analytical estimate requires no cascade sampling at all.

In summary, under various settings and across datasets with diffusion probabilities either synthetically assigned or learned from real action logs, VG-App generally performs well, confirming that it is an effective and robust solution to the MCA problem.

6 Related Work

Graphs have been widely used to represent the relationships of entities in many areas [33, 34, 46, 47, 50, 61–65]. The *misinformation mitigation* (MM) problem is an important problem in graph analysis. Budak et al. [11] first formulate the MM problem under the CIC model. They establish the submodularity of the objective function and exploit this property to design greedy-based algorithms with a $(1 - 1/e - \epsilon)$-approximation. Subsequently, numerous follow-up studies have developed sampling-based algorithms to enhance the efficiency and scalability of MM on large-scale social networks, while preserving the approximation guarantees [41, 42, 44, 52–54]. Moreover, several studies investigate variants of MM, employing alternative strategies such as blocking nodes [56–58, 66] or edges [29, 30, 45] in the original graph to counter influence spread. Different from the above lines of work that aim to minimize the influence by identifying the truth seed set or modifying

the graph structure, this paper explores a new direction that centers on contribution analysis under a *given* mitigation result.

Influence maximization (IM) is a counterpart problem of MM, which aims to identify a seed set that maximizes the influence spread. It has attracted great attention due to its important applications in scenarios such as viral marketing [21, 26]. Kempe et al. [28] first formulate and investigate the IM problem, introducing two classical diffusion models: the independent cascade (IC) and linear threshold (LT) models. Since then, a plethora of heuristics [14–16, 22, 27, 37] and approximation algorithms [10, 23–25, 36, 48, 49, 51] are proposed to reduce the computational overhead.

Recently, Zhu et al. [68] conduct the first study to derive the influence contributions of influencers in social advertising, given the result of an advertising campaign. This task shares a similar objective with ours, in that both aim to quantify the contributions of campaign initiators in the information diffusion process. However, the approximation algorithm in [68] is not applicable to our problem. To explain, [68] adopts the IC model, where only a single campaign is propagated, whereas our study employs the CIC model, in which both truth and misinformation campaigns diffuse simultaneously. As a result, to generate cascades that match the given campaign result, [68] only needs to ensure that each activated node is reachable from an influencer. In contrast, our problem additionally requires considering the specific campaign that activates each node and the tie-breaking mechanism during the diffusion process, making it substantially more complex and non-trivial to handle.

7 Conclusion and Future Work

In this paper, we begin by introducing a contribution metric and establishing its theoretical soundness by connecting it to the Shapley value. We then formulate the mitigation contribution allocation (MCA) problem, which seeks to compute the expected contribution of each influencer conditioned on the given campaign result. To address this, we first propose VG-Sampling for sample generation. On this basis, we develop VG-App, an efficient approximation algorithm with provable accuracy guarantees for MCA. Extensive empirical studies on real-world datasets demonstrate that VG-App achieves accurate approximations while maintaining high efficiency.

Despite these positive results, sustainable misinformation mitigation calls for further investigation along two fronts. On one hand, the CIC model and equal-share allocation adopted in our framework may not always fully capture real-world complexity, and more expressive formulations are worth investigating. Richer diffusion models such as TCIC [41] can be adopted to capture finer-grained dynamics. Meanwhile, the equal-share rule can be refined to account for the varying persuasiveness of different senders (e.g., by assigning larger edge-level weights to authoritative senders), or the decaying influence of late arrivals (where senders delivering the truth after the first reach still receive credit, but with diminishing shares as the delay grows). Notably, VG-App readily accommodates such refinements, as the contribution metric enters only as a modular per-sample scoring step. On the other hand, broader incentive schemes are worth exploring. In practice, to encourage participation, OSN providers often offer multiple schemes in parallel, allowing influencers to choose the one that best aligns with their preferences. Beyond the result-based scheme studied in this paper, the expectation-based scheme, where rewards are determined upfront by each influencer's expected contribution across all possible mitigation outcomes, is also a common option and deserves future study.

Acknowledgments

This work is partially supported by ARC LP210301046, DP240101322, DP260100689. Yanping Wu is the corresponding author.

References

[1] 2026. Online Appendix and Code for "Approximating Contribution in Misinformation Mitigation". https://github.com/wjh0116/MCA.

[2] J Stacy Adams. 1963. Towards an understanding of inequity. *The journal of abnormal and social psychology* 67, 5 (1963), 422.

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

[4] Kursad Asdemir, Nanda Kumar, and Varghese S Jacob. 2012. Pricing models for online advertising: CPM vs. CPC. *Information Systems Research* (2012).

[5] Çigdem Aslay, Francesco Bonchi, Laks V. S. Lakshmanan, and Wei Lu. 2017. Revenue Maximization in Incentivized Social Advertising. *Proc. VLDB Endow.* 10, 11 (2017), 1238–1249.

[6] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. 2006. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 44–54.

[7] Nicola Barbieri and Francesco Bonchi. 2014. Influence Maximization with Viral Product Design. In *SDM*. SIAM, 55–63.

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

[9] 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.

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

[11] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. 2011. Limiting the spread of misinformation in social networks. In *WWW*. ACM, 665–674.

[12] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. 2011. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*. 665–674.

[13] Wei Chen, Carlos Castillo, and Laks VS Lakshmanan. 2013. *Information and influence propagation in social networks*. Morgan & Claypool Publishers.

[14] 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.

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

[16] 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.

[17] CTV News. 2022. Feds spent more than $600K hiring influencers in 2021. CTV News. https://www.ctvnews.ca/politics/article/feds-spent-more-than-600k-hiring-influencers-in-2021/.

[18] Leucio Antonio Cutillo, Mark Manulis, and Thorsten Strufe. 2010. Security and privacy in online social networks. In *Handbook of social network technologies and applications*. Springer, 497–522.

[19] Mickey Djuric. 2024. Ottawa keeps spending on influencers. Liberals say it's about stemming disinformation. The Canadian Press. https://www.timescolonist.com/national-news/ottawa-keeps-spending-on-influencers-liberals-say-its-about-stemming-disinformation-9132050.

[20] 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.

[21] Pedro M. Domingos and Matthew Richardson. 2001. Mining the network value of customers. In *KDD*. ACM, 57–66.

[22] 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.

[23] 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.

[24] Qintian Guo, Sibo Wang, Zhewei Wei, Wenqing Lin, and Jing Tang. 2022. Influence Maximization Revisited: Efficient Sampling with Bound Tightened. *ACM Trans. Database Syst.* 47, 3 (2022), 12:1–12:45.

[25] 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.

[26] Raghuram Iyengar, Christophe Van den Bulte, and Thomas W. Valente. 2011. Opinion Leadership and Social Contagion in New Product Diffusion. *Mark. Sci.* 30, 2 (2011), 195–212.

[27] Kyomin Jung, Wooram Heo, and Wei Chen. 2012. IRIE: Scalable and Robust Influence Maximization in Social Networks. In *ICDM*. IEEE Computer Society, 918–923.

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

[29] Elias Boutros Khalil, Bistra Dilkina, and Le Song. 2014. Scalable diffusion-aware optimization of network topology. In *KDD*. ACM, 1226–1235.
[30] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. 2008. Solving the Contamination Minimization Problem on Networks for the Linear Threshold Model. In *PRICAI (Lecture Notes in Computer Science, Vol. 5351)*. Springer, 977–984.
[31] Edward P Lazear. 2000. Performance pay and productivity. *American Economic Review* (2000).
[32] Jure Leskovec and Julian Mcauley. 2012. Learning to discover social circles in ego networks. *Advances in neural information processing systems* 25 (2012).
[33] Fan Li, Xiaoyang Wang, Chen Chen, and Wenjie Zhang. 2026. Anchor-guided Hypergraph Condensation with Dual-level Discrimination. *arXiv preprint arXiv:2605.10001* (2026).
[34] Fan Li, Xiaoyang Wang, Wenjie Zhang, Ying Zhang, and Xuemin Lin. 2026. DHG-Bench: A Comprehensive Benchmark for Deep Hypergraph Learning. In *The Fourteenth International Conference on Learning Representations*. https://openreview.net/forum?id=lhsb1ChUDF
[35] Taylor Lorenz. 2021. How an 'influencer army' is fighting vaccine lies. San Francisco Examiner. https://www.sfexaminer.com/archives/how-an-influencer-army-is-fighting-vaccine-lies/article_02efe9d9-148d-5b6a-a745-492afe266d9b.html.
[36] 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.
[37] Naoto Ohsaka, Takuya Akiba, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2014. Fast and Accurate Influence Maximization on Large Networks with Pruned Monte-Carlo Simulations. In *AAAI*. AAAI Press, 138–144.
[38] Canice Prendergast. 1999. The provision of incentives in firms. *Journal of economic literature* (1999).
[39] Jon Roozenbeek, Claudia R Schneider, Sarah Dryhurst, John Kerr, Alexandra LJ Freeman, Gabriel Recchia, Anne Marthe Van Der Bles, and Sander Van Der Linden. 2020. Susceptibility to misinformation about COVID-19 around the world. *Royal Society open science* 7, 10 (2020), 201199.
[40] Lloyd S Shapley et al. 1953. A value for n-person games. (1953).
[41] 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.
[42] Michael Simpson, Venkatesh Srinivasan, and Alex Thomo. 2020. Reverse Prevention Sampling for Misinformation Mitigation in Social Networks. In *ICDT (LIPIcs, Vol. 155)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 24:1–24:18.
[43] Michael Simpson, Venkatesh Srinivasan, and Alex Thomo. 2020. Reverse Prevention Sampling for Misinformation Mitigation in Social Networks. In *23rd International Conference on Database Theory*.
[44] Chonggang Song, Wynne Hsu, and Mong-Li Lee. 2017. Temporal Influence Blocking: Minimizing the Effect of Misinformation in Social Networks. In *ICDE*. IEEE Computer Society, 847–858.
[45] Lichao Sun, Xiaobin Rui, and Wei Chen. 2023. Scalable Adversarial Attack Algorithms on Influence Maximization. In *WSDM*. ACM, 760–768.
[46] Xingyu Tan, Shiyuan Liu, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, Liming Zhu, and Wenjie Zhang. 2026. Weaving Multi-Source Evidence for Biomedical Reasoning: The BioMedHop Benchmark and BioWeave Framework. *arXiv preprint arXiv:2606.16211* (2026).
[47] Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, Liming Zhu, and Wenjie Zhang. 2026. PrivGemo: Privacy-Preserving Dual-Tower Graph Retrieval for Empowering LLM Reasoning with Memory Augmentation. *arXiv preprint arXiv:2601.08739* (2026).
[48] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online Processing Algorithms for Influence Maximization. In *SIGMOD Conference*. ACM, 991–1005.
[49] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence Maximization in Near-Linear Time: A Martingale Approach. In *SIGMOD Conference*. ACM, 1539–1554.
[50] Yao Tang, Jinghao Wang, Chen Chen, Xiaoyang Wang, Wenjie Zhang, and Ying Zhang. 2024. Mitigate the Damage of Rumor on Susceptible Group. In *Australasian Database Conference*. Springer, 389–402.
[51] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: near-optimal time complexity meets practical efficiency. In *SIGMOD Conference*. ACM, 75–86.
[52] Guangmo Amo Tong and Ding-Zhu Du. 2019. Beyond Uniform Reverse Sampling: A Hybrid Sampling Technique for Misinformation Prevention. In *INFOCOM*. IEEE, 1711–1719.
[53] Guangmo Amo Tong, Ding-Zhu Du, and Weili Wu. 2018. On Misinformation Containment in Online Social Networks. In *NeurIPS*. 339–349.
[54] Guangmo Amo Tong, Weili Wu, Ling Guo, Deying Li, Cong Liu, Bin Liu, and Ding-Zhu Du. 2017. An efficient randomized algorithm for rumor blocking in online social networks. In *INFOCOM*. IEEE, 1–9.
[55] Jinghao Wang, Yanping Wu, Xiaoyang Wang, Chen Chen, Ying Zhang, and Lu Qin. 2025. Effective influence maximization with priority. In *Proceedings of the ACM on Web Conference 2025*. 4673–4683.

[56] 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.
[57] 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 (2025), 1–27.
[58] 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.
[59] Xiaoyang Wang, Ying Zhang, Wenjie Zhang, and Xuemin Lin. 2016. Efficient distance-aware influence maximization in geo-social networks. *IEEE transactions on knowledge and data engineering* 29, 3 (2016), 599–612.
[60] Xiaoyang Wang, Ying Zhang, Wenjie Zhang, Xuemin Lin, and Chen Chen. 2016. Bring order into the samples: A novel scalable method for influence maximization. *IEEE Transactions on Knowledge and Data Engineering* 29, 2 (2016), 243–256.
[61] 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.
[62] 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.
[63] Yanping Wu, Renjie Sun, Xiaoyang Wang, Ying Zhang, Lu Qin, Wenjie Zhang, and Xuemin Lin. 2025. Efficient k-Plex Mining in Temporal Graphs. *IEEE Trans. Knowl. Data Eng.* 37, 12 (2025), 7105–7119.
[64] 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.
[65] 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* 27, 1 (2024), 6.
[66] 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.
[67] Jaewon Yang and Jure Leskovec. 2012. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD workshop on mining data semantics*. 1–8.
[68] Yuqing Zhu, Jing Tang, Xueyan Tang, and Lei Chen. 2021. Analysis of Influence Contribution in Social Advertising. *Proc. VLDB Endow.* 15, 2 (2021), 348–360.
[69] 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 17 January 2026; revised 19 May 2026; accepted 12 June 2026