# From Competition to Complementarity: Comparative Influence Diffusion and Maximization

Wei Lu
University of British Columbia
Vancouver, B.C., Canada
welu@cs.ubc.ca

Wei Chen
Microsoft Research
Beijing, China
weic@microsoft.com

Laks V.S. Lakshmanan
University of British Columbia
Vancouver, B.C., Canada
laks@cs.ubc.ca

## ABSTRACT

Influence maximization is a well-studied problem that asks for a small set of influential users from a social network, such that by targeting them as early adopters, the expected total adoption through influence cascades over the network is maximized. However, almost all prior work focuses on cascades of a single propagating entity or purely-competitive entities. In this work, we propose the *Comparative Independent Cascade* (Com-IC) model that covers the full spectrum of entity interactions from competition to complementarity. In Com-IC, users' adoption decisions depend not only on edge-level information propagation, but also on a node-level automaton whose behavior is governed by a set of model parameters, enabling our model to capture not only competition, but also complementarity, to *any possible degree*. We study two natural optimization problems, *Self Influence Maximization* and *Complementary Influence Maximization*, in a novel setting with complementary entities. Both problems are NP-hard, and we devise efficient and effective approximation algorithms via non-trivial techniques based on reverse-reachable sets and a novel "sandwich approximation" strategy. The applicability of both techniques extends beyond our model and problems. Our experiments show that the proposed algorithms consistently outperform intuitive baselines on four real-world social networks, often by a significant margin. In addition, we learn model parameters from real user action logs.

## 1. INTRODUCTION

Online social networks are ubiquitous and play an essential role in our daily life. Fueled by popular applications such as viral marketing, there has been extensive research in influence and information propagation in social networks, from both theoretical and practical points of view. A key computational problem in this field is *influence maximization*, which asks to identify a small set of $k$ influential users (also known as *seeds*) from a given social network, such that by targeting them as early adopters of a new technology, product, or opinion, the expected number of total adoptions triggered by social influence cascade (or, propagation) is maximized [9, 16]. The dynamics of an influence cascade are typically governed by a *stochastic diffusion model*, which specifies how adoptions propagate from one user to another in the network.

Most existing work focuses on two types of diffusion models — *single-entity models* and *pure-competition models*. A single-entity model has only one propagating entity for social network users to adopt: the classic Independent Cascade (IC) and Linear Thresholds (LT) models [16] belong to this category. These models, however, ignore complex social interactions involving multiple propagating entities. Considerable work has been done to extend IC and LT models to study competitive influence maximization, but almost all models assume that the propagating entities are in pure competition and users adopt at most one of them [2, 4, 6–8, 14, 17, 22].

In reality, the relationship between different propagating entities is certainly more general than pure competition. In fact, consumer theories in economics have two well-known notions: *substitute goods* and *complementary goods* [23]. Substitute goods are similar ones that compete, and can be purchased, one in place of the other, e.g., smartphones of various brands. Complementary goods are those that tend to be purchased together, e.g,. iPhone and its accessories, computer hardware and software, etc. There are also varying *degrees* of substitutability and complementarity: buying a product could lessen the probability of buying the other without necessarily eliminating it; similarly, buying a product could boost the probability of buying another to any degree. Pure competition only corresponds to the special case of perfect substitute goods.

The limitation of pure-competition models can be exposed by the following example. Consider a viral marketing campaign featuring iPhone 6 and Apple Watch. It is vital to recognize the fact that Apple Watch generally needs an iPhone to be usable, and iPhone's user experience can be greatly enhanced by a pairing Apple Watch (see, e.g., *http://bit.ly/1GOqesc*). Clearly none of the pure-competition models is suitable for this campaign because they do not even allow users to adopt both the phone and the watch! This motivates us to design a more powerful, expressive, yet reasonably tractable model that captures not only competition, but also complementarity, and to any possible degrees associated with these notions.

To this end, we propose the *Comparative Independent Cascade* model, or Com-IC for short, which, unlike most existing diffusion models, consists of two critical components that work jointly to govern the dynamics of diffusions: edge-level information propagation and a *Node-Level Automaton* (NLA) that ultimately makes adoption decisions based on a set of model parameters, known as the *Global Adoption Probabilities* (GAPs). Of these, edge-level propagation is similar to the propagation captured by the classical IC and LT models, but only controls *information awareness*. The NLA is a novel feature and is unique to our proposal. Indeed, the term "comparative" comes from the fact that once a user is aware, via edge-level propagation, of multiple products, intuitively


*Proceedings of the VLDB Endowment,* Vol. 9, No. 2

she makes a comparison between them by "running" her NLA. Notice that "comparative" subsumes "competitive" and "complementary" as special cases. In theory, the Com-IC model is able to accommodate any number of propagating entities (items) and cover the entire spectrum from competition to complementarity between pairs of items, reflected by the values of GAPs. In this work, as the first step toward comparative influence diffusion and viral marketing, we focus on the case of two items. At any time, w.r.t. any item $\mathcal{A}$, a user in the network is in one of the following four states: $\mathcal{A}$-*idle*, $\mathcal{A}$-*suspended*, $\mathcal{A}$-*rejected*, or $\mathcal{A}$-*adopted*. The NLA sets out probabilistic transition rules between states, and different GAPs are applied based on *a given user's state w.r.t. the other item $\mathcal{B}$ and the relationship between $\mathcal{A}$ and $\mathcal{B}$*. Intuitively, competition (complementarity) is modeled as reduced probability (resp., increased probability) of adopting the second item after the first item is already adopted. After a user adopts an item, she propagates this information to her neighbors in the network, making them aware of the item. The neighbor may adopt the item with a certain probability, as governed by her NLA.

We then define two novel optimization problems for two complementary items $\mathcal{A}$ and $\mathcal{B}$. Our first problem, *Self Influence Maximization (*SELFINFMAX*)*, asks for $k$ seeds for $\mathcal{A}$ such that given a fixed set of $\mathcal{B}$-seeds, the expected number of $\mathcal{A}$-adopted nodes is maximized. The second one, *Complementary Influence Maximization (*COMPINFMAX*)*, considers the flip side of SELFINFMAX: given a fixed set of $\mathcal{A}$-seeds, find a set of $k$ seeds for $\mathcal{B}$ such that the expected increase in $\mathcal{A}$-adopted nodes thanks to $\mathcal{B}$ is maximized. *To the best of our knowledge, we are the first to systematically study influence maximization for complementary items.*

We show that both problems are NP-hard under Com-IC. Moreover, two important properties, *submodularity* and *monotonicity* (see §2), which would allow a greedy approximation algorithm frequently used for influence maximization, do not hold in unrestricted Com-IC model. Even when we restrict Com-IC to mutual complementarity, submodularity still does not hold in general.

To circumvent the aforementioned difficulties, we first show that submodularity holds for a subset of the complementary parameter space. We then make a non-trivial extension to the *Reverse-Reachable Set* (RR-set) techniques [3, 24, 25], originally proposed for influence maximization with single-entity models, to obtain effective and efficient approximation solutions to both SELFINFMAX and COMPINFMAX. Next, we propose a novel *Sandwich Approximation* (SA) strategy which, for a given non-submodular set function, provides an upper bound function and/or a lower bound function, and uses them to obtain data-dependent approximation solutions w.r.t. the original function. We further note that both techniques are applicable to a larger context beyond the model and problems studied in this paper: for RR-sets, we provide a new definition and general sufficient conditions not covered by [3, 24, 25] that apply to a large family of influence diffusion models, while SA applies to the maximization of any non-submodular functions that are upper- and/or lower-bounded by submodular functions.

In experiments, we first learn GAPs from user action logs from two social networking sites – Flixster.com and Douban.com. We demonstrate that our approximation algorithms based on RR-sets and SA techniques consistently outperform several intuitive baselines, typically by a significant margin on real-world networks.

To summarize, we make the following contributions:

• We propose the Com-IC model to characterize influence diffusion dynamics of products with arbitrary degree of competition or complementarity, and identify a subset of the parameter space under which submodularity and monotonicity of influence spread hold, paving the way for approximation algorithms (§3 and §5).

• We propose two novel problems – Self Influence Maximization and Complementary Influence Maximization – for complementary products under the Com-IC model (§4).

• We show that both problems are NP-hard, and devise efficient and effective approximation solutions by non-trivial extensions to RR-set techniques and by proposing Sandwich Approximation, both having applicability beyond this work (§6).

• We conduct empirical evaluations on four real-world social networks and demonstrate the superiority of our algorithms over intuitive baselines, and also propose a methodology for learning global adoption probabilities for the Com-IC model from user action logs of social networking sites (§7).

For lack of space, we omit some technical proofs and additional results; they are presented in the full version of this paper [1].

## 2. BACKGROUND & RELATED WORK

Given a graph $G = (V, E, p)$ where $p : E \to [0, 1]$ specifies pairwise influence probabilities (or weights) between nodes, and $k \in \mathbb{Z}_+$, the *influence maximization* problem asks to find a set $S \subseteq V$ of $k$ seeds, activating which leads to the maximum expected number of active nodes (denoted $\sigma(S)$) [16]. Under both IC and LT models, this problem is NP-hard; Chen et al. [10, 11] showed computing $\sigma(S)$ exactly for any $S \subseteq V$ is #P-hard. Fortunately, $\sigma(\cdot)$ is a *submodular* and *monotone* function of $S$ for both IC and LT, which allows a simple greedy algorithm with an approximation factor of $1 - 1/e - \epsilon$, for any $\epsilon > 0$ [16, 21]. A set function $f : 2^U \to \mathbb{R}_{\geq 0}$ is submodular if for any $S \subseteq T \subseteq U$ and any $x \in U \setminus T$, $f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - T(S)$, and monotone if $f(S) \leq f(T)$ whenever $S \subseteq T \subseteq U$. Tang et al. [24, 25] proposed new randomized approximation algorithms which are orders of magnitude faster than the original greedy algorithms in [16].

In competitive influence maximization [2, 4, 6–8, 14, 17, 22] (also surveyed in [9]), a common thread is the focus on pure competition, which only allows users to adopt at most one product or opinion. Most works are from the follower's perspective [2, 6, 7, 14], i.e., given competitor's seeds, how to maximize one's own spread, or minimize the competitor's spread. [17] aims to maximize the total influence spread of all competitors while ensuring fair allocation.

For viral marketing with non-competing items, Datta et al. [12] studied influence maximization with items whose propagations are independent. Narayanam et al. [20] studied a setting with two sets of products, where a product can be adopted by a node only when it has already adopted a corresponding product in the other set. Their model extends LT. We depart by defining a significantly more powerful and expressive model in Com-IC, compared to theirs which only covers the special case of *perfect complementarity*.

Myers and Leskovec analyzed Twitter data to study the effects of different cascades on users and predicted the likelihood of a user adopting a piece of information given the cascades to which the user was previously exposed [19]. McAuley et al. used logistic regression to learn substitute/complementary relationships between products from user reviews [18]. Both studies focus on data analysis and behavior prediction and do not provide diffusion modeling for competing and complementary items, nor do they study the influence maximization problem in this context.

## 3. COMPARATIVE IC MODEL

**Review of Classical IC Model.** In the IC model [16], there is just one entity (e.g., idea or product) being propagated through the network. An instance of the model has a directed graph $G = (V, E, p)$ where $p : E \to [0, 1]$, and a seed set $S \subset V$. For convenience, we use $p_{u,v}$ for $p(u, v)$. At time step 0, the seeds are *active* and

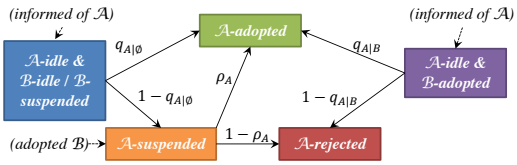



**Figure 1: Com-IC model: Node-level automaton for product $\mathcal{A}$.**

all other nodes are *inactive*. Propagation proceeds in discrete time steps. At time $t$, every node $u$ that became active at $t-1$ makes one attempt to activate each of its inactive out-neighbors $v$. This can be seen as node $u$ "testing" if the edge $(u, v)$ is "live" or "blocked". The out-neighbor $v$ becomes active at $t$ iff the edge is live. The propagation ends when no new nodes become active.

**Key differences from IC model.** In the *Comparative IC model* (Com-IC), there are at least two products. For ease of exposition, we focus on just two products $\mathcal{A}$ and $\mathcal{B}$ below. Each node can be in any of the states {*idle*, *suspended*, *adopted*, *rejected*} w.r.t. *each* of the products. All nodes are initially in the joint state of ($\mathcal{A}$-idle, $\mathcal{B}$-idle). One of the biggest differences between Com-IC and IC is the separation of information diffusion (edge-level) and the actual adoption decisions (node-level). Edges only control the information that flows to a node: e.g., when $u$ adopts a product, its out-neighbor $v$ may be informed of this fact. Once that happens, $v$ uses its own node level automaton (NLA) to decide which state to transit to. This depends on $v$'s current state w.r.t. the two products as well as parameters corresponding to the state transition probabilities of the NLA, namely the Global Adoption Probabilities, defined below.

A concise representation of the NLA is in Figure 1. Each state is indicated by the label. The state diagram is self-explanatory. E.g., with probability $q_{\mathcal{A}|\emptyset}$, a node transits from a state where it's $\mathcal{A}$-idle to $\mathcal{A}$-adopted, regardless of whether it was $\mathcal{B}$-idle or $\mathcal{B}$-suspended.

From the $\mathcal{A}$-suspended state, it transits to $\mathcal{A}$-adopted w.p. $\rho_{\mathcal{A}}$ and to $\mathcal{A}$-rejected w.p. $1-\rho_{\mathcal{A}}$. The probability $\rho_{\mathcal{A}}$, called *reconsideration probability*, as well as the reconsideration process will be explained below. Note that in a Com-IC diffusion process defined below, not all joint state is reachable from the initial ($\mathcal{A}$-idle, $\mathcal{B}$-idle) state, e.g., ($\mathcal{A}$-idle, $\mathcal{B}$-rejected). Since all unreachable states are irrelevant to adoptions, they are negligible (details in [1]).

**Global Adoption Probability (GAP).** GAPs, consisting of four parameters $\mathbf{Q} = (q_{\mathcal{A}|\emptyset}, q_{\mathcal{A}|\mathcal{B}}, q_{\mathcal{B}|\emptyset}, q_{\mathcal{B}|\mathcal{A}}) \in [0,1]^4$ are important parameters of the NLA which decide the likelihood of adoptions after a user is informed of an item. $q_{\mathcal{A}|\emptyset}$ is the probability that a user adopts $\mathcal{A}$ given that she is informed of $\mathcal{A}$ but not $\mathcal{B}$-adopted, and $q_{\mathcal{A}|\mathcal{B}}$ is the probability that a user adopts $\mathcal{A}$ given that she is already $\mathcal{B}$-adopted. A similar interpretation applies to $q_{\mathcal{B}|\emptyset}$ and $q_{\mathcal{B}|\mathcal{A}}$.

Intuitively, GAPs reflect the overall popularity of products and how they are perceived by the entire market. They are considered aggregate estimates and hence are not user specific in our model. We provide further justifications at the end of this section and describe a way to learn GAPs from user action log data in §7.

GAPs enable Com-IC to model competition and complementarity, to arbitrary degrees. We say that $\mathcal{A}$ *competes* with $\mathcal{B}$ iff $q_{\mathcal{B}|\mathcal{A}} \leq q_{\mathcal{B}|\emptyset}$. Similarly, $\mathcal{A}$ *complements* $\mathcal{B}$ iff $q_{\mathcal{B}|\mathcal{A}} \geq q_{\mathcal{B}|\emptyset}$. We include the special case of $q_{\mathcal{B}|\mathcal{A}} = q_{\mathcal{B}|\emptyset}$ in both cases above for convenience of stating our technical results, and it actually means that the propagation of $\mathcal{B}$ is completely independent of $\mathcal{A}$ (*cf.* Lemma 3). Competition and complementarity in the other direction are similar. The degree of competition and complementarity is determined by the difference between the two relevant GAPs, i.e., $|q_{\mathcal{B}|\mathcal{A}} - q_{\mathcal{B}|\emptyset}|$ and $|q_{\mathcal{A}|\mathcal{B}} - q_{\mathcal{A}|\emptyset}|$. For convenience, we use $\mathbf{Q}^+$ to refer to any set of GAPs representing *mutual complementarity*: $(q_{\mathcal{A}|\emptyset} \leq q_{\mathcal{A}|\mathcal{B}}) \wedge (q_{\mathcal{B}|\emptyset} \leq q_{\mathcal{B}|\mathcal{A}})$, and similarly, $\mathbf{Q}^-$ for GAPs representing *mutual competition*: $(q_{\mathcal{A}|\emptyset} \geq q_{\mathcal{A}|\mathcal{B}}) \wedge (q_{\mathcal{B}|\emptyset} \geq q_{\mathcal{B}|\mathcal{A}})$.

**Global Iteration.** At every time step $t \geq 1$, for all nodes that became $\mathcal{A}$- or $\mathcal{B}$-adopted at $t-1$, their outgoing edges are tested for transition (1 below). After that, for each node $v$ that has at least one in-neighbor (with a live edge) becoming $\mathcal{A}$- and/or $\mathcal{B}$-adopted at $t-1$, $v$ is tested for possible state transition (2-4 below).

1. **Edge transition.** For an untested edge $(u, v)$, flip a biased coin independently: $(u, v)$ is *live* w.p. $p_{u,v}$ and *blocked* w.p. $1-p_{u,v}$. Each edge is tested at most once in the entire diffusion process.
2. **Node tie-breaking.** Consider a node $v$ to be tested at time $t$. Generate a random permutation $\pi$ of $v$'s in-neighbors (with live edges) that adopted at least one product at $t-1$. Then, test $v$ with each such in-neighbor $u$ and $u$'s adopted item ($\mathcal{A}$ and/or $\mathcal{B}$) following $\pi$. If there is a $w \in N^-(v)$ adopting both $\mathcal{A}$ and $\mathcal{B}$, then test both products, following their order of adoption by $w$.
3. **Node adoption.** Consider the case of testing an $\mathcal{A}$-idle node $v$ for adopting $\mathcal{A}$ (Figure 1). If $v$ is *not* $\mathcal{B}$-adopted, then w.p. $q_{\mathcal{A}|\emptyset}$, it becomes $\mathcal{A}$-adopted and w.p. $1-q_{\mathcal{A}|\emptyset}$ it becomes $\mathcal{A}$-suspended. If $v$ is $\mathcal{B}$-adopted, then w.p. $q_{\mathcal{A}|\mathcal{B}}$, it becomes $\mathcal{A}$-adopted and w.p. $1-q_{\mathcal{A}|\mathcal{B}}$ it becomes $\mathcal{A}$-rejected. The case of adopting $\mathcal{B}$ is symmetric.
4. **Node reconsideration.** Consider an $\mathcal{A}$-suspended node $v$ that just adopts $\mathcal{B}$ at time $t$. Define $\rho_{\mathcal{A}} = \max\{q_{\mathcal{A}|\mathcal{B}} - q_{\mathcal{A}|\emptyset}, 0\}/(1 - q_{\mathcal{A}|\emptyset})$. Then, $v$ *reconsiders* to become $\mathcal{A}$-adopted w.p. $\rho_{\mathcal{A}}$, or $\mathcal{A}$-rejected w.p. $1-\rho_{\mathcal{A}}$. The case of reconsidering $\mathcal{B}$ is symmetric.

**Figure 2: Com-IC model: diffusion dynamics**

**Diffusion dynamics.** Let $G = (V, E, p)$ be a directed social graph with pairwise influence probabilities. Let $S_{\mathcal{A}}, S_{\mathcal{B}} \subset V$ be the seed sets for $\mathcal{A}$ and $\mathcal{B}$. Influence diffusion under Com-IC proceeds in discrete time steps. Initially, every node is $\mathcal{A}$-idle and $\mathcal{B}$-idle. At time step 0, every $u \in S_{\mathcal{A}}$ becomes $\mathcal{A}$-adopted and every $u \in S_{\mathcal{B}}$ becomes $\mathcal{B}$-adopted[1]. If $u \in S_{\mathcal{A}} \cap S_{\mathcal{B}}$, we randomly decide the order of $u$ adopting $\mathcal{A}$ and $\mathcal{B}$ with a fair coin. For ease of understanding, we describe the rest of the diffusion process in a modular way in Figure 2. We use $N^+(v)$ and $N^-(v)$ to denote the set of out-neighbors and in-neighbors of $v$, respectively.

We draw special attention to *tie-breaking* and *reconsideration*. Tie-breaking is used when a node's in-neighbors adopt different products and try to inform the node at the same step. Node reconsideration concerns the situation that a node $v$ did not adopt $\mathcal{A}$ initially but later after adopting $\mathcal{B}$ it may reconsider adopting $\mathcal{A}$: when $\mathcal{B}$ competes with $\mathcal{A}$ ($q_{\mathcal{A}|\emptyset} \geq q_{\mathcal{A}|\mathcal{B}}$), $v$ will not reconsider adopting $\mathcal{A}$, but when $\mathcal{B}$ complements $\mathcal{A}$ (specifically, $q_{\mathcal{A}|\emptyset} < q_{\mathcal{A}|\mathcal{B}}$), $v$ will reconsider adopting $\mathcal{A}$. In the latter case, the probability of adopting $\mathcal{A}$, $\rho_{\mathcal{A}}$, is defined in such a way that the overall probability of adopting $\mathcal{A}$ is equal to $q_{\mathcal{A}|\mathcal{B}}$ (since $q_{\mathcal{A}|\mathcal{B}} = q_{\mathcal{A}|\emptyset} + (1 - q_{\mathcal{A}|\emptyset})\rho_{\mathcal{A}}$).

**Design Considerations.** The design of Com-IC not only draws on the essential elements from a classical diffusion model (IC) stemming from mathematical sociology, but also closes a gap between theory and practice, in which diffusions typically do not occur just for one product or with just one mode of pure competition. With GAPs in the NLA, the model can characterize *any* possible relationship between two propagating entities: competition, complementarity, and any degree associated with them. GAPs are fully capable of handling asymmetric relationship between products. E.g., an Apple Watch ($\mathcal{A}$) is complemented more by an iPhone ($\mathcal{B}$) than the other way round: many functionalities of the watch are not usable without a pairing iPhone, but an iPhone is totally functional without a

[1] No generality is lost in assuming seeds adopt an item without testing the NLA: for every $v \in V$, we can create two dummy nodes $v_{\mathcal{A}}, v_{\mathcal{B}}$ and edges $(v_{\mathcal{A}}, v)$ and $(v_{\mathcal{B}}, v)$ with $p_{v_{\mathcal{A}},v} = p_{v_{\mathcal{B}},v} = 1$. Requiring seeds to go through NLA is equivalent to constraining that $\mathcal{A}$-seeds ($\mathcal{B}$-seeds) be selected from all $v_{\mathcal{A}}$'s (resp. $v_{\mathcal{B}}$'s).

watch. This asymmetric complementarity can be expressed by any GAPs satisfying $(q_{\mathcal{A}|\mathcal{B}} - q_{\mathcal{A}|\emptyset}) > (q_{\mathcal{B}|\mathcal{A}} - q_{\mathcal{B}|\emptyset}) \geq 0$. Furthermore, introducing NLA with GAPs and separating the propagation of product information from actual adoptions reflects Kalish's famous characterization of new product adoption [15]: customers go through two stages – *awareness* followed by *actual adoption*. In Kalish's theory, product awareness is propagated through word-of-mouth effects; after an individual becomes aware, she would decide whether to adopt the item based on other considerations. Edges in the network can be seen as information channels from one user to another. Once the channel is open (live), it remains so. This modeling choice is reasonable as competitive goods are typically of the same kind and complementary goods tend to be adopted together.

We remark that Com-IC encompasses previously-studied single-entity and pure-competition models as special cases. When $q_{\mathcal{A}|\emptyset} = q_{\mathcal{B}|\emptyset} = 1$ and $q_{\mathcal{A}|\mathcal{B}} = q_{\mathcal{B}|\mathcal{A}} = 0$, Com-IC reduces to the (purely) Competitive Independent Cascade model [9]. If, in addition, $q_{\mathcal{B}|\emptyset}$ is 0, the model further reduces to the classic IC model.

# 4. FORMAL PROBLEM STATEMENTS

Many interesting optimization problems can be formulated under the expressive Com-IC model. In this work, we focus on influence maximization with *complementary propagating entities*, since competitive viral marketing has been studied extensively (see §2). In what follows, we propose two problems. The first one, *Self Influence Maximization* (SELFINFMAX), is a natural extension to the classical influence maximization problem [16]. The second one is the novel *Complementary Influence Maximization* (COMPINFMAX), where the objective is to maximize complementary effects (or "boost" the expected number of adoptions) by selecting the best seeds of a complementing good.

Given the seed sets $S_{\mathcal{A}}, S_{\mathcal{B}}$, we first define $\sigma_{\mathcal{A}}(S_{\mathcal{A}}, S_{\mathcal{B}})$ and $\sigma_{\mathcal{B}}(S_{\mathcal{A}}, S_{\mathcal{B}})$ to be the expected number of $\mathcal{A}$-adopted and $\mathcal{B}$-adopted nodes, respectively under the Com-IC model. Clearly, both $\sigma_{\mathcal{A}}$ and $\sigma_{\mathcal{B}}$ are real-valued bi-set functions mapping $2^V \times 2^V$ to $[0, |V|]$, for any fixed $\mathbf{Q}$. Unless otherwise specified, GAPs are not considered as arguments to $\sigma_{\mathcal{A}}$ and $\sigma_{\mathcal{B}}$ as $\mathbf{Q}$ is constant in a given instance of Com-IC. Also, for simplicity, we may refer to $\sigma_{\mathcal{A}}(\cdot,\cdot)$ as $\mathcal{A}$-spread and $\sigma_{\mathcal{B}}(\cdot,\cdot)$ as $\mathcal{B}$-spread. The following two problems are defined in terms of $\mathcal{A}$-spread, without loss of generality.

PROBLEM 1 (SELFINFMAX). *Given a directed graph $G = (V, E, p)$ with pairwise influence probabilities, $\mathcal{B}$-seed set $S_{\mathcal{B}} \subset V$, a cardinality constraint $k$, and a set of GAPs $\mathbf{Q}^+$, find an $\mathcal{A}$-seed set $S^*_{\mathcal{A}} \subset V$ of size $k$, such that the* expected number of $\mathcal{A}$-adopted nodes *is maximized under Com-IC: $S^*_{\mathcal{A}} \in \arg\max_{T \subseteq V, |T|=k} \sigma_{\mathcal{A}}(T, S_{\mathcal{B}})$.*

SELFINFMAX is obviously NP-hard, as it subsumes INFMAX under the classic IC model when $S_{\mathcal{B}} = \emptyset$ and $q_{\mathcal{A}|\emptyset} = q_{\mathcal{A}|\mathcal{B}} = 1$. By a similar argument, it is #P-hard to compute the exact value of $\sigma_{\mathcal{A}}(S_{\mathcal{A}}, S_{\mathcal{B}})$ and $\sigma_{\mathcal{B}}(S_{\mathcal{A}}, S_{\mathcal{B}})$ for any given $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$.

PROBLEM 2 (COMPINFMAX). *Given a directed graph $G = (V, E, p)$ with pairwise influence probabilities, $\mathcal{A}$-seed set $S_{\mathcal{A}} \subset V$, a cardinality constraint $k$, and a set of GAPs $\mathbf{Q}^+$, find a $\mathcal{B}$-seed set $S^*_{\mathcal{B}} \subseteq V$ of size $k$ such that the* expected increase (boost) in $\mathcal{A}$-adopted nodes *is maximized under Com-IC: $S^*_{\mathcal{B}} \in \arg\max_{T \subseteq V, |T|=k} [\sigma_{\mathcal{A}}(S_{\mathcal{A}}, T) - \sigma_{\mathcal{A}}(S_{\mathcal{A}}, \emptyset)]$.*

THEOREM 1. *COMPINFMAX is NP-hard.*

From the formulation of COMPINFMAX, we can intuitively see that the placement of $\mathcal{B}$-seeds will be heavily dependent on the existing $\mathcal{A}$-seeds. For example, if $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ are in two different connected components of the graph, then evidently the boost is zero. In contrast, if they are rather close and can influence roughly the same region in the graph, the boost is likely to be high. Indeed, for the special case where $q_{\mathcal{B}|\emptyset} = 1$ and $k \geq |S_{\mathcal{A}}|$, directly "copying" $\mathcal{A}$-seeds to be $\mathcal{B}$-seeds will give the optimal boost. However, this in no way diminishes the value of this problem, as the NP-hardness remains as long as $q_{\mathcal{B}|\emptyset} \neq 1$ *or* $k < |S_{\mathcal{A}}|$.

THEOREM 2. *For COMPINFMAX, when $q_{\mathcal{B}|\emptyset} = 1$ and $k \geq |S_{\mathcal{A}}|$, we can solve the problem optimally by setting $S^*_{\mathcal{B}}$ to be $S_{\mathcal{A}} \cup X$, where $X$ is an arbitrary set in $V \setminus S_{\mathcal{A}}$ with size $k - |S_{\mathcal{A}}|$, that is $\sigma_{\mathcal{A}}(S_{\mathcal{A}}, S_{\mathcal{A}} \cup X) = \max_{T \subseteq V, |T|=k} \sigma_{\mathcal{A}}(S_{\mathcal{A}}, T)$.*

# 5. PROPERTIES OF Com-IC

Since neither of SELFINFMAX and COMPINFMAX can be solved in PTIME unless P = NP, we explore approximation algorithms by studying submodularity and monotonity for Com-IC, which may pave the way for designing approximation algorithms. Note that $\sigma_{\mathcal{A}}$ is a bi-set function taking arguments $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$, so we analyze the properties w.r.t. each of the two arguments. As appropriate, we refer to the properties of $\sigma_{\mathcal{A}}$ w.r.t. $S_{\mathcal{A}}$ ($S_{\mathcal{B}}$) as *self-monotonicity* (resp., *cross-monotonicity*) and *self-submodularity* (resp., *cross-submodularity*).

## 5.1 An Equivalent Possible World Model

To facilitate a better understanding of Com-IC and our analysis on submodularity, we define a *Possible World (PW) model* that provides an equivalent view of the Com-IC model. Given a graph $G = (V, E, p)$ and a diffusion model, a possible world consists of a *deterministic graph* sampled from a probability distribution over all subgraphs of $G$. For Com-IC, we also need some variables for each node to fix the outcomes of random events in relation to the NLA (adoption, tie-breaking, and reconsideration), so that influence cascade is fully deterministic in a single possible world.

**Generative Rules.** Retain each edge $(u, v) \in E$ w.p. $p_{u,v}$ (*live* edge) and drop it w.p. $1 - p_{u,v}$ (*blocked* edge). This generates a possible world $W$ with $G_W = (V, E_W)$, $E_W$ being the set of live edges. Next, $\forall$ node $v \in V$: $(i)$ choose "thresholds" $\alpha^{v,W}_{\mathcal{A}}$ and $\alpha^{v,W}_{\mathcal{B}}$ independently and uniformly at random from $[0, 1]$, for comparison with GAPs in adoption decisions (when $W$ is clear from context, we write $\alpha^v_{\mathcal{A}}$ and $\alpha^v_{\mathcal{B}}$); $(ii)$ generate a random permutation $\pi_v$ of all in-neighbors $u \in N^-(v)$ (for tie-breaking); $(iii)$ sample a discrete value $\tau_v \in \{\mathcal{A}, \mathcal{B}\}$, where each value has a probability of 0.5 (used for tie-breaking in case $v$ is a seed of both $\mathcal{A}$ and $\mathcal{B}$).

**Deterministic cascade in a PW.** At time step 0, nodes in $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ first become $\mathcal{A}$-adopted and $\mathcal{B}$-adopted, respectively (ties, if any, are broken based on $\tau_v$). Then, iteratively for each step $t \geq 1$, a node $v$ becomes "reachable" by $\mathcal{A}$ at time step $t$ if $t$ is the length of a shortest path from any seed $u \in S_{\mathcal{A}}$ to $v$ consisting entirely of live edges and $\mathcal{A}$-adopted nodes. Node $v$ then becomes $\mathcal{A}$-adopted at step $t$ if $\alpha^v_{\mathcal{A}} \leq x$, where $x = q_{\mathcal{A}|\emptyset}$ if $v$ is not $\mathcal{B}$-adopted, otherwise $x = q_{\mathcal{A}|\mathcal{B}}$. For re-consideration, suppose $v$ just becomes $\mathcal{B}$-adopted at step $t$, while being $\mathcal{A}$-suspended (i.e., $v$ was reachable by $\mathcal{A}$ before $t$ steps but $\alpha^v_{\mathcal{A}} > q_{\mathcal{A}|\emptyset}$). Then, $v$ adopts $\mathcal{A}$ if $\alpha^v_{\mathcal{A}} \leq q_{\mathcal{A}|\mathcal{B}}$. The reachability and reconsideration tests of $\mathcal{B}$ are symmetric. For *tie-breaking*, if $v$ is reached by both $\mathcal{A}$ and $\mathcal{B}$ at $t$, the permutation $\pi_v$ is used to determine the order in which $\mathcal{A}$ and $\mathcal{B}$ are considered. In addition, if $v$ is reached by $\mathcal{A}$ and $\mathcal{B}$ from the same in-neighbor, e.g., $u$, then the informing order follows the order in which $u$ adopts $\mathcal{A}$ and $\mathcal{B}$.

The following lemma establishes the equivalence between this possible world model and Com-IC. This allows us to analyze monotonicity and submodularity using the PW model only.

LEMMA 1. *For any fixed $\mathcal{A}$-seed set $S_\mathcal{A}$ and $\mathcal{B}$-seed set $S_\mathcal{B}$, the joint distributions of the sets of $\mathcal{A}$-adopted nodes and $\mathcal{B}$-adopted nodes obtained $(i)$ by running a Com-IC diffusion from $S_\mathcal{A}$ and $S_\mathcal{B}$ and $(ii)$ by randomly sampling a possible world $W$ and running a deterministic cascade from $S_\mathcal{A}$ and $S_\mathcal{B}$ in $W$, are the same.*

**Remarks on Monotonicity.** It turns out that when $\mathcal{A}$ competes with $\mathcal{B}$ while $\mathcal{B}$ complements $A$, monotonicity does not hold in general (see [1] for counter-examples). However, these cases are rather unnatural, and thus we next focus on mutual competition ($\mathbf{Q}^-$) and mutural complementary cases ($\mathbf{Q}^+$), for which we can show self- and cross-monotonicity do hold.

THEOREM 3. *For any fixed $\mathcal{B}$-seed set $S_\mathcal{B}$, $\sigma_\mathcal{A}(S_\mathcal{A}, S_\mathcal{B})$ is monotonically increasing in $S_\mathcal{A}$ for any set of GAPs in $\mathbf{Q}^+$ and $\mathbf{Q}^-$. Also, $\sigma_\mathcal{A}(S_\mathcal{A}, S_\mathcal{B})$ is monotonically increasing in $S_\mathcal{B}$ for any GAPs in $\mathbf{Q}^+$, and monotonically decreasing in $S_\mathcal{B}$ for any $\mathbf{Q}^-$.*

## 5.2 Submodularity in Complementary Setting

Next, we analyze self-submodularity and cross-submodularity for mutual complementary cases ($\mathbf{Q}^+$) that has direct impact on SELFINFMAX and COMPINFMAX. The analysis for $\mathbf{Q}^-$ is in [1].

For self-submodularity, we show that it is satisfied in the case of *"one-way complementarity"*, i.e., $\mathcal{B}$ complements $\mathcal{A}$ ($q_{\mathcal{A}|\emptyset} \leq q_{\mathcal{A}|\mathcal{B}}$), but $\mathcal{A}$ does not affect $\mathcal{B}$ ($q_{\mathcal{B}|\emptyset} = q_{\mathcal{B}|\mathcal{A}}$), or vise versa (Theorem 4). We will also show the $\sigma_\mathcal{A}$ is cross-submodular in $S_\mathcal{B}$ when $q_{\mathcal{B}|\mathcal{A}} = 1$ (Theorem 5). However, both properties are not satisfied in general (see [1] for counter-examples). We give two useful lemmas first, and thanks to Lemma 2 below, we may assume w.l.o.g. that tie-breaking always favors $\mathcal{A}$ in complementary cases.

LEMMA 2. *Consider any Com-IC instance with $\mathbf{Q}^+$. Given fixed $\mathcal{A}$- and $\mathcal{B}$-seed sets, for all nodes $v \in V$, all permutations of $v$'s in-neighbors are equivalent in determining if $v$ becomes $\mathcal{A}$-adopted and $\mathcal{B}$-adopted, and thus the tie-breaking rule is not needed for mutual complementary case.*

LEMMA 3. *In the Com-IC model, if $\mathcal{B}$ is indifferent to $\mathcal{A}$ (i.e., $q_{\mathcal{B}|\mathcal{A}} = q_{\mathcal{B}|\emptyset}$), then for any fixed $\mathcal{B}$ seed set $S_\mathcal{B}$, the probability distribution over sets of $\mathcal{B}$-adopted nodes is independent of $\mathcal{A}$-seed set. Symmetrically, the probability distribution over sets of $\mathcal{A}$-adopted nodes is also independent of $\mathcal{B}$-seed set if $\mathcal{A}$ is indifferent to $\mathcal{B}$.*

THEOREM 4. *For any instance of Com-IC model with $q_{\mathcal{A}|\emptyset} \leq q_{\mathcal{A}|\mathcal{B}}$ and $q_{\mathcal{B}|\emptyset} = q_{\mathcal{B}|\mathcal{A}}$, $(i)$. $\sigma_\mathcal{A}$ is* self-submodular *w.r.t. $\mathcal{A}$ seed set $S_\mathcal{A}$, for any fixed $\mathcal{B}$-seed set $S_\mathcal{B}$. $(ii)$. $\sigma_\mathcal{B}$ is* self-submodular *w.r.t. $\mathcal{B}$ seed set $S_\mathcal{B}$ and is independent of $\mathcal{A}$-seed set $S_\mathcal{A}$.*

PROOF. First, $(ii)$ holds trivially. By Lemma 3, $\mathcal{A}$ does not affect $\mathcal{B}$'s diffusion in any sense. Thus, $\sigma_\mathcal{B}(S_\mathcal{A}, S_\mathcal{B}) = \sigma_\mathcal{B}(\emptyset, S_\mathcal{B})$. It can be shown that the function $\sigma_\mathcal{B}(\emptyset, S_\mathcal{B})$ is both monotone and submodular w.r.t. $S_\mathcal{B}$, for any $q_{\mathcal{B}|\emptyset}$, through a straightforward extension to the proof of Theorem 2.2 in Kempe et al. [16].

For $(i)$, first we fix a possible world $W$ and a $\mathcal{B}$-seed set $S_\mathcal{B}$. Let $\Phi^W_\mathcal{A}(S_\mathcal{A})$ be the set of $\mathcal{A}$-adopted nodes in possible world $W$ with $\mathcal{A}$-seed set $S_\mathcal{A}$ ($S_\mathcal{B}$ omitted when it is clear from the context). Consider two sets $S \subseteq T \subseteq V$, some node $u \in V \setminus T$, and finally a node $v \in \Phi^W_\mathcal{A}(T \cup \{u\}) \setminus \Phi^W_\mathcal{A}(T)$. There must exist a live-edge path $P_\mathcal{A}$ from $T \cup \{u\}$ consisting entirely of $\mathcal{A}$-adopted nodes. We denote by $w_0 \in T \cup \{u\}$ the origin of $P_\mathcal{A}$.

We first prove a key claim: $P_\mathcal{A}$ remains $\mathcal{A}$-adopted when $S_\mathcal{A} = \{w_0\}$. Consider any node $w_i \in P_\mathcal{A}$. In this possible world, if $\alpha^{w_i}_\mathcal{A} \leq q_{\mathcal{A}|\emptyset}$, then regardless of the diffusion of $\mathcal{B}$, $w_i$ will adopt $\mathcal{A}$ as long as its predecessor $w_{i-1}$ adopts $\mathcal{A}$. If $q_{\mathcal{A}|\emptyset} < \alpha^{w_i}_\mathcal{A} \leq q_{\mathcal{A}|\mathcal{B}}$, then there must also be a live-edge path $P_\mathcal{B}$ from $S_\mathcal{B}$ to $w_i$ that consists entirely of $\mathcal{B}$-adopted nodes, and it boosts $w_i$ to adopt $\mathcal{A}$. Since $q_{\mathcal{B}|\emptyset} = q_{\mathcal{B}|\mathcal{A}}$, $\mathcal{A}$ has no effect on $\mathcal{B}$-propagation (Lemma 3), and $P_\mathcal{B}$ always exists and all nodes on $P_\mathcal{B}$ would still be $\mathcal{B}$-adopted through $S_\mathcal{B}$ (fixed) irrespective of $\mathcal{A}$-seeds. Thus, $P_\mathcal{B}$ always boosts $w_i$ to adopt $\mathcal{A}$ as long as $w_{i-1}$ is $\mathcal{A}$-adopted. Hence, the claim holds by a simple induction on $P_\mathcal{A}$ starting from $w_0$.

Then, it is easy to see $w_0 = u$. Suppose otherwise, then $w_0 \in T$ must be true. By the claim above and self-monotonicity of $\sigma_\mathcal{A}$ (Theorem 3), $v \in \Phi^W_\mathcal{A}(\{w_0\})$ implies $w \in \Phi^W_\mathcal{A}(T)$, a contradiction. Therefore, we have $v \notin \Phi^W_\mathcal{A}(S)$ and $v \in \Phi^W_\mathcal{A}(S \cup \{u\})$. This by definition implies $|\Phi^W_\mathcal{A}(\cdot)|$ is submodular for any $W$ and $S_\mathcal{B}$, which is sufficient to show that $\sigma_\mathcal{A}(S_\mathcal{A}, S_\mathcal{B})$ is submodular in $S_\mathcal{A}$. □

THEOREM 5. *In any instance of Com-IC with mutual complementarity $\mathbf{Q}^+$, $\sigma_\mathcal{A}$ is* cross-submodular *w.r.t. $\mathcal{B}$-seed set $S_\mathcal{B}$, for any fixed $A$-seed set, as long as $q_{\mathcal{B}|\mathcal{A}} = 1$.*

PROOF. We first fix an $\mathcal{A}$-seed set $S_\mathcal{A}$. Consider any possible world $W$. Let $\Psi^W_\mathcal{A}(S_\mathcal{B})$ be the set of $\mathcal{A}$-adopted nodes in $W$ with $\mathcal{B}$ seed-set $S_\mathcal{B}$ (and $\mathcal{A}$-seed set $S_\mathcal{A}$). Consider $\mathcal{B}$-seed sets $S \subseteq T \subseteq V$ and another $\mathcal{B}$-seed $u \in V \setminus T$. It suffices to show that for any $v \in \Psi^W_\mathcal{A}(T \cup \{u\}) \setminus \Psi^W_\mathcal{A}(T)$, we have $v \in \Psi^W_\mathcal{A}(S \cup \{u\}) \setminus \Psi^W_\mathcal{A}(S)$.

Let an *$\mathcal{A}$-path* be a live-edge path from some $\mathcal{A}$-seed such that all nodes on the path adopt $\mathcal{A}$, and $\mathcal{B}$-path is defined symmetrically. If a node $w$ has $\alpha^w_\mathcal{A} \leq q_{\mathcal{A}|\emptyset}$, we say that $w$ is *$\mathcal{A}$-ready*, meaning that $w$ is ready for $\mathcal{A}$ and will adopt $\mathcal{A}$ if it is informed of $\mathcal{A}$, regardless of its status on $\mathcal{B}$. We say a path from $S_\mathcal{A}$ is an *$\mathcal{A}$-ready path* if all nodes on the path (except the starting $\mathcal{A}$-seed) are $\mathcal{A}$-ready. It is clear that all nodes on an $\mathcal{A}$-ready path would always adopt $\mathcal{A}$ regardless of $\mathcal{B}$-seeds. We define $\mathcal{B}$-ready nodes and paths symmetrically. We can show the following claim (proofs in [1]).

CLAIM 1. *On any $\mathcal{A}$-path $P_\mathcal{A}$, if some node $w$ adopts $\mathcal{B}$ and all nodes before $w$ on $P_\mathcal{A}$ are $\mathcal{A}$-ready, then every node following $w$ on $P_\mathcal{A}$ adopts both $\mathcal{A}$ and $\mathcal{B}$, regardless of the actual $\mathcal{B}$-seed set.*

Now consider the case of $S_\mathcal{B} = T \cup \{u\}$ first. Since $v \in \Psi^W_\mathcal{A}(T \cup \{u\})$, there must be an $\mathcal{A}$-path $P_\mathcal{A}$ from some node $w_0 \in S_\mathcal{A}$ to $v$. If path $P_\mathcal{A}$ is $\mathcal{A}$-ready, then regardless of $\mathcal{B}$ seeds, all nodes on $P_\mathcal{A}$ would always be $\mathcal{A}$-adopted, but this contradicts the assumption that $v \notin \Psi^W_\mathcal{A}(T)$. Therefore, there exists some node $w$ that is not $\mathcal{A}$-ready, i.e., $q_{\mathcal{A}|\emptyset} < \alpha^w_\mathcal{A} \leq q_{\mathcal{A}|\mathcal{B}}$. Let $w$ be the first non-$\mathcal{A}$-ready node on path $P_\mathcal{A}$. Then $w$ must have adopted $\mathcal{B}$ to help it adopt $\mathcal{A}$, and $\alpha^w_\mathcal{B} \leq q_{\mathcal{B}|\emptyset}$. We can show the following key claim.

CLAIM 2. *There is a $\mathcal{B}$-path $P_\mathcal{B}$ from some $\mathcal{B}$-seed $x_0 \in T \cup \{u\}$ to $w$, such that even if $x_0$ is the only $\mathcal{B}$-seed, $w$ still adopts $\mathcal{B}$.*

With the key Claim 2, the rest of the proof follows the standard argument as in the other proofs. In particular, since even when $x_0$ is the only $\mathcal{B}$-seed, $w$ can still be $\mathcal{B}$-adopted, then by Claim 1, $v$ would be $\mathcal{A}$-adopted in this case. Thus we know that $x_0$ must be $u$, because otherwise it contradicts our assumption that $v \notin \Psi^W_\mathcal{A}(T)$ (also relying on the cross-monotonicity proof made for Theorem 3). Then again by the cross-monotonicity, we know that $v \in \Psi^W_\mathcal{A}(S \cup \{u\})$, but $v \notin \Psi^W_\mathcal{A}(S)$. This completes our proof. □

# 6. APPROXIMATION ALGORITHMS

We first review the state-of-the-art in influence maximization and then derive a general framework (§6.1) to obtain approximation algorithms for SELFINFMAX (§6.2) and COMPINFMAX (§6.3).

**TIM algorithm.** For influence maximization, Tang et al. [25] proposed the *Two-phase Influence Maximization (TIM)* algorithm that produces a $(1-1/e-\epsilon)$-approximation with at least $1-|V|^{-\ell}$ probability in $O((k+\ell)(|E|+|V|)\log|V|/\epsilon^2)$ expected running time. It is based on the concept of *Reverse-Reachable sets (RR-sets)* [3],

and applies to the Triggering model [16] that generalizes both IC and LT. TIM is *orders of magnitude faster than greedy algorithm with Monte Carlo simulations [16], while still giving approximation solutions with high probability*. Recently they propose a new improvement [24], which significantly reduces the number of RR-sets generated using martingale analysis. To tackle SELFINFMAX and COMPINFMAX, we primarily focus on the challenging task of correctly generating RR-sets in Com-IC and other more general models, which is orthogonal to the contributions of [24]. Hereafter we focus on the framework of [25]. Due to the much more complex dynamics involved in Com-IC, adapting TIM to solve SELFINFMAX and COMPINFMAX is far from trivial, as we shall show.

**Reverse-Reachable Sets.** In a deterministic (directed) graph $G' = (V', E')$, for a fixed $v \in V'$, all nodes that can reach $v$ form an *RR-set* rooted at $v$ [3], denoted $R(v)$. A *random* RR set encapsulates two levels of randomness: $(i)$ a "root" node $v$ is randomly chosen from the graph, and $(ii)$ a deterministic graph is sampled according to a certain probabilistic rule that retains a subset of edges from the graph. E.g., for the IC model, each edge $(u, v) \in E$ is removed w.p. $(1-p_{u,v})$, independently. TIM first computes a lower bound on the optimal solution value and uses this bound to derive the number of random RR-sets to be sampled, denoted $\theta$. To guarantee approximation solutions, $\theta$ must satisfy:

$$\theta \geq \epsilon^{-2}(8+2\epsilon)|V| \cdot \frac{\ell \log |V| + \log \binom{|V|}{k} + \log 2}{OPT_k}, \quad (1)$$

where $OPT_k$ is the optimal influence spread achievable amongst all size-$k$ sets, and $\epsilon$ represents the trade-off between efficiency and quality: a smaller $\epsilon$ implies more RR-sets (longer running time), but gives a better approximation factor. The approximation guarantee of TIM relies on a key result from [3], re-stated here:

PROPOSITION 1 (LEMMA 9 IN [25]). *Fix a set $S \subseteq V$ and a node $v \in V$. Under the Triggering model, let $\rho_1$ be the probability that $S$ activates $v$ in a cascade, and $\rho_2$ be the probability that $S$ overlaps with a random RR-set $R(v)$ rooted at $v$. Then, $\rho_1 = \rho_2$.*

## 6.1 A General Solution Framework

We use Possible World (PW) models to generalize the theory in [3, 25]. For a generic stochastic diffusion model $M$, an *equivalent* PW model $M'$ is a model that specifies a distribution over $\mathcal{W}$, the set of all possible worlds, where influence diffusion in each possible world in $\mathcal{W}$ is deterministic. Further, given a seed set (or two seed sets $S_\mathcal{A}$ and $S_\mathcal{B}$ as in Com-IC), the distribution of the sets of active nodes (or $\mathcal{A}$- and $\mathcal{B}$-adopted nodes in Com-IC) in $M$ is the same as the corresponding distribution in $M'$. Then, we define a generalized concept of *RR-set* through the PW model:

DEFINITION 1 (GENERAL RR-SET). *For each possible world $W \in \mathcal{W}$ and a given node $v$ (a.k.a. root), the* reverse reachable set (RR-set) of $v$ in $W$, *denoted by $R_W(v)$, consists of all nodes $u$ such that the singleton set $\{u\}$ would activate $v$ in $W$. A* random *RR-set of $v$ is a set $R_W(v)$ where $W$ is randomly sampled from $\mathcal{W}$ using the probability distribution given in $M'$.*

It is easy to see that Definition 1 encompasses the RR-set definition in [3, 25] for IC, LT, and Triggering models as special cases. For the entire solution framework to work, the key property that RR-sets need to satisfy is the following:

DEFINITION 2 (ACTIVATION EQUIVALENCE PROPERTY). *Let $M$ be a stochastic diffusion model and $M'$ be its equivalent possible world model. Let $G = (V, E, p)$ be a graph. Then, RR-sets have the* Activation Equivalence Property *if for any fixed $S \subseteq V$ and any fixed $v \in V$, the probability that $S$ activates $v$ according to $M$ is the same as the probability that $S$ overlaps with a random RR-set generated from $v$ in a possible world in $M'$.*

As shown in [25], the entire correctness and complexity analysis is based on the above property, and in fact in their latest improvement [24], they directly use this property as the definiton of general RR-sets. Proposition 1 shows that the activation equivalence property holds for the triggering model. We now provide *a more general sufficient condition for the activation equivalence property to hold* (Lemma 5), which gives *concrete conditions on when the RR-set based framework would work*. More specifically, we show that for any diffusion model $M$, if there is an equivalent PW model $M'$ of which all possible worlds satisfy the following two properties, then RR-sets have the activation equivalence property.

**Possible World Properties.** *(P1):* Given two seed sets $S \subseteq T$, if a node $v$ can be activated by $S$ in a possible world $W$, then $v$ shall also be activated by $T$ in $W$. *(P2):* If a node $v$ can be activated by $S$ in a possible world $W$, then there exists $u \in S$ such that the singleton seed set $\{u\}$ can also activate $v$ in $W$. In fact, *(P1)* and *(P2)* are equivalent to monotonicity and submodularity, as we formally state below.

LEMMA 4. *Let $W$ be a fixed possible world. Let $f_{v,W}(S)$ be an indicator function that takes on* 1 *if $S$ can activate $v$ in $W$, and* 0 *otherwise. Then, $f_{v,W}(\cdot)$ is* monotone *and* submodular *for all $v \in V$ if and only if both (P1) and (P2) are satisfied in $W$.*

LEMMA 5. *Let $M$ be a stochastic diffusion model and $M'$ be its equivalent possible world model. If $M'$ satisfies Properties (P1) and (P2), then the RR-sets as defined in Definition 1 have the activation equivalence property as in Definition 2.*

Comparing with directly using the activation equivalence property as the RR-set definition in [24], our RR-set definition provides a more concrete way of constructing RR-sets, and our Lemmas 4 and 5 provide general conditions under which such constructions can ensure algorithm correctness. Algorithm 1, GeneralTIM, outlines a general solution framework based on RR-sets and TIM. It provides a probabilistic approximation guarantee for any diffusion models that satisfy *(P1)* and *(P2)*. Note that the estimation of a lower bound $LB$ of $OPT_k$ (line 1) is orthogonal to our contributions and we refer the reader to [25] for details. Finally, we have:

THEOREM 6. *Suppose for a stochastic diffusion model $M$ with an equivalent PW model $M'$, that for every possible world $W$ and every $v \in V$, the indicator function $f_{v,W}$ is monotone & submodular. Then for influence maximization under $M$ with graph $G = (V, E, p)$ and seed set size $k$,* GeneralTIM *(Algorithm 1) applied on the general RR-sets (Definition 1) returns a $(1 - 1/e - \epsilon)$-approximate solution with at least $1 - |V|^{-\ell}$ probability.*

Theorem 6 follows from Lemmas 4 and 5, and the fact that all theoretical analysis of TIM relies only on the Chernoff bound and the activation equivalence property, "without relying on any other results specific to the IC model" [25]. Next, we describe how to generate RR-sets correctly for SELFINFMAX and COMPINFMAX under Com-IC (line 3 of Algorithm 1), which is much more complicated than IC/LT models [25]. We will first focus on submodular settings for SELFINFMAX (Theorem 4) and COMPINFMAX (Theorem 5). In §6.4, we propose *Sandwich Approximation* to handle general $\mathbf{Q}^+$ where submodularity does *not* hold.

## 6.2 Generating RR-Sets for SELFINFMAX

We present two algorithms, RR-SIM and RR-SIM+, for generating random RR-sets per Definition 1. The overall algorithm for SELFINFMAX can be obtained by plugging RR-SIM or RR-SIM+ into GeneralTIM (Algorithm 1).

**Algorithm 1:** GeneralTIM ($G = (V, E, p)$, $k$, $\epsilon$, $\ell$)

$LB \leftarrow$ lower bound of $OPT_k$ estimated by method in [25]
compute $\theta$ using Eq. (1) with $LB$ replacing $OPT_k$
$\mathcal{R} \leftarrow$ generate $\theta$ random RR-sets according to Definition 1 *// for* SELFINFMAX, *use* RR-SIM *or* RR-SIM+; *for* COMPINFMAX, *use* RR-CIM
**for** $i = 1$ to $k$ **do**
  $v_i \leftarrow$ the node appearing in the most RR-sets in $\mathcal{R}$
  $S \leftarrow S \cup \{v_i\}$ *// $S$ was initialized as $\emptyset$*
  remove all RR-sets in which $v_i$ appears
**return** $S$ as the seed set

According to Definition 1, for SELFINFMAX, the RR-set of a root $v$ in a possible world $W$, $R_W(v)$, is the set of nodes $u$ such that if $u$ is the only $\mathcal{A}$-seed, $v$ would be $\mathcal{A}$-adopted in $W$, given any fixed $\mathcal{B}$-seed set $S_\mathcal{B}$. By Theorems 3 and 4 (whose proofs indeed show that the indicator function $f_{v,W}(S)$ is monotone and submodular), along with Lemmas 4 and 5, we know that RR-sets following Definition 1 have the activation equivalence property. We now focus on how to construct RR-sets following Definition 1. Recall that in Com-IC, adoption decisions for $\mathcal{A}$ are based on a number of factors such as whether $v$ is reachable via a live-edge path from $S_\mathcal{A}$ and its state w.r.t. $\mathcal{B}$ when reached by $\mathcal{A}$. Note that $q_{\mathcal{B}|\emptyset} = q_{\mathcal{B}|\mathcal{A}}$ implies that $\mathcal{B}$-diffusion is independent of $\mathcal{A}$ (Lemma 3). Our algorithms take advantage of this fact, by first revealing node states w.r.t. $\mathcal{B}$, which gives a sound basis for generating RR-sets for $\mathcal{A}$.

### 6.2.1 The RR-SIM Algorithm

Conceptually, RR-SIM (Algorithm 2) proceeds in three phases. Phase I samples a possible world according to §5.1 (omitted from the pseudo-code). Phase II is a *forward labeling* process from the input $\mathcal{B}$-seed set $S_\mathcal{B}$ (lines 2 to 7): a node $v$ becomes $\mathcal{B}$-adopted if $\alpha_\mathcal{B}^{v,W} \leq q_{\mathcal{B}|\emptyset}$ and $v$ is reachable from $S_\mathcal{B}$ via a path consisting entirely of live edges and $\mathcal{B}$-adopted nodes. In Phase III (lines 8 to 15), we randomly select a node $v$ and generate RR-set $R_W(v)$ by running a Breadth-First Search (BFS) backwards (following incoming edges). Note that the RR-set generation for IC and LT models [25] is essentially a simpler version of Phase III.

**Backward BFS.** Given $W$, an RR-set $R_W(v)$ includes all nodes explored in the following backward BFS procedure. Initially, we enqueue $v$ into a FIFO queue $Q$. We repeatedly dequeue a node $u$ from $Q$ for processing until the queue is empty.

*Case 1: $u$ is $\mathcal{B}$-adopted.* There are two sub-cases: $(i)$. If $\alpha_\mathcal{A}^u \leq q_{\mathcal{A}|\mathcal{B}}$, then $u$ is able to transit from $\mathcal{A}$-informed to $\mathcal{A}$-adopted. Thus, we continue to examine $u$'s in-neighbors. For all unexplored $w \in N^-(u)$, if edge $(w, u)$ is live, then enqueue $w$; $(ii)$. If $\alpha_\mathcal{A}^u > q_{\mathcal{A}|\mathcal{B}}$, then $u$ cannot transit from $\mathcal{A}$-informed to $\mathcal{A}$-adopted, and thus $u$ has to be an $\mathcal{A}$ seed to become $\mathcal{A}$-adopted. In this case, $u$'s in-neighbors will not be examined.

*Case 2: $u$ is not $\mathcal{B}$-adopted.* Similarly, if $\alpha_\mathcal{A}^u \leq q_{\mathcal{A}|\emptyset}$, perform actions as in $1(i)$; otherwise perform actions as in $1(ii)$.

THEOREM 7. *Under one-way complementarity ($q_{\mathcal{A}|\emptyset} \leq q_{\mathcal{A}|\mathcal{B}}$ and $q_{\mathcal{B}|\emptyset} = q_{\mathcal{B}|\mathcal{A}}$), the RR-sets generated by the* RR-SIM *algorithm satisfy Definition 1 for the* SELFINFMAX *problem. As a result, Theorem 6 applies to* GeneralTIM *with* RR-SIM *in this case.*

**Lazy sampling.** For RR-SIM to work, it is *not* necessary to sample all edge- and node-level variables (i.e., the entire possible world) up front, as the forward labeling and backward BFS are unlikely to reach the whole graph. Hence, we can simply reveal edge and node states on demand ("lazy sampling"), based on the principle of deferred decisions. In light of this observation, the following improvements are made to RR-SIM. First, the first phase is simply skipped. Second, in Phase II, edge states and $\alpha$-values are sampled as the forward labeling from $S_\mathcal{B}$ goes on. We record the outcomes, as it is possible to encounter certain edges and nodes again in phase (III). Next, for Phase III, consider any node $u$ dequeued from $Q$. We need to perform an additional check on every incoming edge $(w, u)$. If $(w, u)$ has already been tested live in Phase II, then we just enqueue $w$. Otherwise, we first sample its live/blocked status, and enqueue $w$ if it is live, Algorithm 2 provides the pseudo-code for RR-SIM, where sampling is assumed to be done whenever we need to check the status of an edge or the $\alpha$-values of a node.

**Algorithm 2:** RR-SIM ($G = (V, E)$, $v$, $S_\mathcal{B}$)

1 create an empty FIFO queue $Q$ and empty set $R$
2 enqueue all nodes in $S_\mathcal{B}$ into $Q$ *// start forward labeling*
3 **while** $Q$ is not empty **do**
4  $u \leftarrow Q$.dequeue() and mark $u$ as $B$-adopted
5  **foreach** $v \in N^+(u)$ such that $(u, v)$ is live **do**
6  **if** $\alpha_\mathcal{B}^{v,W} \leq q_{\mathcal{B}|\emptyset} \wedge v$ is not visited **then**
7  $Q$.enqueue($v$) *// also mark $v$ as visited*
8 clear $Q$, and then enqueue $v$ *// start backward BFS*
9 **while** $Q$ is not empty **do**
10  $u \leftarrow Q$.dequeue()
11  $R \leftarrow R \cup \{u\}$
12  **if** ($u$ is $\mathcal{B}$-adopted $\wedge\ \alpha_\mathcal{A}^{u,W} \leq q_{\mathcal{A}|\mathcal{B}}$) $\vee$ ($u$ is not $\mathcal{B}$-adopted $\wedge\ \alpha_\mathcal{A}^{u,W} \leq q_{\mathcal{A}|\emptyset}$) **then**
13  **foreach** $w \in N^-(u)$ such that $(w, u)$ is live **do**
14  **if** $w$ is not visited **then**
15  $Q$.enqueue($w$) *// also mark $w$ visited*
16 **return** $R$ as the RR-set

**Expected time complexity.** For the entire seed selection (Algorithm 1 with RR-SIM) to guarantee approximate solutions, we must estimate a lower bound $LB$ of $OPT_k$ and use it to derive the minimum number of RR-sets required, defined as $\theta$ in Eq. (1). In expectation, the algorithm runs in $O(\theta \cdot EPT)$ time, where $EPT$ is the expected number of edges explored in generating one RR-set. Clearly, $EPT = EPT_F + EPT_B$, where $EPT_F$ ($EPT_B$) is the expected number of edges examined in forward labeling (resp., backward BFS). Thus, we have the following result.

LEMMA 6. *In expectation,* GeneralTIM *with* RR-SIM *runs in* $O\left((k+\ell)(|V|+|E|)\log|V|\,(1+EPT_F/EPT_B)\right)$ *time.*

$EPT_F$ increases when the input $\mathcal{B}$-seed set grows. Intuitively, it is reasonable that a larger $\mathcal{B}$-seed set may have more complementary effect and thus it may take longer time to find the best $\mathcal{A}$-seed set. However, it is possible to reduce $EPT_F$ as described below.

### 6.2.2 The RR-SIM+ Algorithm

The RR-SIM algorithm may incur efficiency loss because some of the work done in forward labeling (Phase II) may not be used in backward BFS (Phase III). E.g., consider an extreme situation where all nodes explored in forward labeling are in a different connected component of the graph than the root $v$ of the RR-set. In this case, forward labeling can be skipped safely and entirely! To leverage this, we propose RR-SIM+ (pseudo-code included in [1]), of which the key idea is to run *two* rounds of backward BFS from the random root $v$. The first round determines the *necessary scope* of forward labeling, while the second one generates the RR-set.

**First backward BFS.** As usual, we create a FIFO queue $Q$ and enqueue the random root $v$. We also sample $\alpha_\mathcal{B}^v$ uniformly at random from $[0, 1]$. Then we repeatedly dequeue a node $u$ until $Q$ is empty: for each incoming edge $(w, u)$, we test its live/blocked status based on probability $p_{w,u}$, independently. If $(w, u)$ is live and $w$ has not been visited before, enqueue $w$ and sample its $\alpha_\mathcal{B}^w$.

**Algorithm 3:** RR-CIM ($G = (V, E)$, $v$, $S_{\mathcal{A}}$)

```
1  conduct forward labeling on G from S_A, cf. Eq. (2)
2  if v is neither A-suspended or A-potential then
3  |  return ∅ as the RR-set
4  Q.enqueue(v) // Q initialized as an empty FIFO queue
5  while Q is not empty do
6  |  u ← Q.dequeue()
7  |  if u is A-suspended then
8  |  |  if u is AB-diffusible then
9  |  |  |  R ← R ∪ {u} // R was initialized as ∅
10 |  |  |  conduct a secondary backward BFS from u via B
   |  |  |  diffusible nodes, and add all explored nodes to R
11 |  |  else R ← R ∪ {u}
12 |  else if u is A-potential then
13 |  |  if u is AB-diffusible then
14 |  |  |  foreach unvisited w ∈ N⁻(u) s.t. (w, u) live do
15 |  |  |  |  Q.enqueue(w); // also mark it visited
16 |  |  else
17 |  |  |  S_f ← nodes visited in a secondary forward BFS
18 |  |  |  S_b ← nodes visited in a secondary backward BFS
19 |  |  |  R ← R ∪ {u} if S_f ∩ S_b contains an A-suspended
   |  |  |  node u_0
20 return R as the RR-set
```

Let $T_1$ be the set of all nodes explored. If $T_1 \cap S_{\mathcal{B}} = \emptyset$, then none of the $\mathcal{B}$-seeds can reach the explored nodes, so that forward labeling can be completely skipped. The above extreme example falls into this case. Otherwise, we run a *residual* forward labeling only from $T_1 \cap S_{\mathcal{B}}$ along the explored nodes in $T_1$: if a node $u \in T_1 \setminus S_{\mathcal{B}}$ is reachable by some $s \in T_1 \cap S_{\mathcal{B}}$ via a live-edge path with all $\mathcal{B}$-adopted nodes, and $\alpha_{\mathcal{B}}^{u,W} \leq q_{\mathcal{B}|\emptyset}$, $u$ becomes $\mathcal{B}$-adopted. Note that it is not guaranteed in theory that this always saves time compared to RR-SIM, since the worst case of RR-SIM+ is that $T_1 \cap S_{\mathcal{B}} = S_{\mathcal{B}}$, which means that the first round is wasted. However, our experimental results §7 indeed show that RR-SIM+ is at least twice faster than RR-SIM on three of the four datasets.

**Second backward BFS.** This round is largely the same as Phase III in RR-SIM, but there is a subtle difference. Suppose we just dequeued a node $u$. It is possible that there exists an incoming edge $(w, u)$ whose status is not determined. This is because we do not enqueue previously visited nodes in BFS. Hence, if in the previous round, $w$ is already visited via an out-neighbor other than $u$, $(w, u)$ would not be tested. Thus, in the current round we shall test $(w, u)$, and decide if $w$ belongs to $R_W(v)$ accordingly. To see RR-SIM+ is equivalent to RR-SIM, it suffices to show that for each node explored in the second backward BFS, its adoption status w.r.t. $\mathcal{B}$ is the same in both algorithms. We prove this fact in [1].

The analysis on expected time complexity is similar: We can show that the expected running time of RR-SIM+ is $O\left((k+\ell)(|V|+|E|)\log|V|\,(1+EPT_{B1}/EPT_{B2})\right)$, where $EPT_{B1}$ ($EPT_{B2}$) is the expected number of edges explored in the first (resp., second) backward BFS. Compared to RR-SIM, $EPT_{B2}$ is the same as $EPT_B$ in RR-SIM, so RR-SIM+ will be faster than RR-SIM if $EPT_{B1} < EPT_F$, i.e., if the first backward BFS plus the residual forward labeling *explores fewer edges*, compared to the full orward labeling in RR-SIM.

## 6.3 Generating RR-Sets for CompInfMax

In CompInfMax, by Definition 1, a node $u$ belongs to an RR-set $R_W(v)$ iff $v$ is not $\mathcal{A}$-adopted without any $\mathcal{B}$-seed, but turns $\mathcal{A}$-adopted when $u$ is the only $\mathcal{B}$-seed. It turns out that constructing RR-sets for CompInfMax following the above definition is significantly more difficult than that for SelfInfMax. This is because when $q_{\mathcal{A}|\emptyset} \leq q_{\mathcal{A}|\mathcal{B}}$ and $q_{\mathcal{B}|\emptyset} \leq q_{\mathcal{B}|\mathcal{A}} = 1$, $\mathcal{A}$ and $\mathcal{B}$ complement each other, and thus a simple forward labeling from the fixed $\mathcal{A}$-seed set, without knowing anything about $\mathcal{B}$, will not be able to determine the $\mathcal{A}$ adoption status of all nodes. This is in contrast to SelfInfMax with one-way complementarity for which $\mathcal{B}$-diffusion is fully independent of $\mathcal{A}$. Thus, when generating RR-sets for CompInfMax, we have to determine more complicated status in a forward labeling process from $\mathcal{A}$-seeds, as shown below.

**Phase I: forward labeling.** The nature of CompInfMax requires us to identify nodes with the potential to be $\mathcal{A}$-adopted with the help of $\mathcal{B}$. To this end, we first conduct a forward search from $S_{\mathcal{A}}$ to label the nodes their status of $\mathcal{A}$. As in RR-SIM, we also employ lazy sampling. The algorithm first enqueues all $\mathcal{A}$-seeds (and labels them $\mathcal{A}$-adopted) into a FIFO queue $Q$. Then we repeatedly dequeue a node $u$ for processing until $Q$ is empty. Let $v$ be an out-neighbor of $u$. Flip a coin with bias $p_{u,v}$ to determine if edge $(u, v)$ is live. If yes, we determine the label of $v$ to be one of the following:

$$\begin{cases} \mathcal{A}\text{-adopted}, & \text{if } u \text{ is } \mathcal{A}\text{-adopted} \wedge \alpha_{\mathcal{A}}^v \leq q_{\mathcal{A}|\emptyset} \\ \mathcal{A}\text{-rejected}, & \text{if } \alpha_{\mathcal{A}}^v > q_{\mathcal{A}|\mathcal{B}}\text{, regardless of } u\text{'s status} \\ \mathcal{A}\text{-suspended}, & \text{if } u \text{ is } \mathcal{A}\text{-adopted} \wedge \alpha_{\mathcal{A}}^v \in (q_{\mathcal{A}|\emptyset}, q_{\mathcal{A}|\mathcal{B}}] \\ \mathcal{A}\text{-potential}, & \text{if } u \text{ is } \mathcal{A}\text{-suspended/potential} \wedge \alpha_{\mathcal{A}}^v \leq q_{\mathcal{A}|\mathcal{B}} \end{cases} \tag{2}$$

Here, $\mathcal{A}$-potential is just a label used for bookkeeping and is not a state. Then node $v$ is added to $Q$ unless it is $\mathcal{A}$-rejected. Note that both $\mathcal{A}$-suspended and $\mathcal{A}$-potential nodes can turn into $\mathcal{A}$-adopted with the complementary effect of $\mathcal{B}$. The main difference is an $\mathcal{A}$-suspended node is informed of $\mathcal{A}$, while an $\mathcal{A}$-potential is not and the informing action must be triggered by $\mathcal{B}$-propagation. Also, unlike a typical BFS, the forward labeling may need to revisit a node: if $u$ is $\mathcal{A}$-adopted (just dequeued) and $v$ is previously labeled $\mathcal{A}$-potential, $v$ should be "promoted" to $\mathcal{A}$-suspended. This occurs when $v$ is first reached by a live-edge path through an $\mathcal{A}$-suspended/potential in-neighbor, but later $v$ is reached by a longer path through an $\mathcal{A}$-adopted in-neighbor.

To facilitate the second phase, we define additional node labels *$\mathcal{AB}$-diffusible* and *$\mathcal{B}$-diffusible*. Node $v$ is $\mathcal{AB}$-diffusible if $v$ can adopt both $\mathcal{A}$ and $\mathcal{B}$ when $v$ is informed about both $\mathcal{A}$ and $\mathcal{B}$; while $v$ is $\mathcal{B}$-diffusible if $v$ can adopt $\mathcal{B}$ when it is informed about $\mathcal{B}$. Accordingly, the technical conditions for them are given below:

$$\begin{cases} \mathcal{AB}\text{-diffusible}, & \text{if } \alpha_{\mathcal{A}}^v \leq q_{\mathcal{A}|\emptyset} \vee \ ((q_{\mathcal{A}|\emptyset} < \alpha_{\mathcal{A}}^v \leq q_{\mathcal{A}|\mathcal{B}}) \wedge (\alpha_{\mathcal{B}}^v \leq q_{\mathcal{B}|\emptyset})) \\ \mathcal{B}\text{-diffusible}, & \text{if } \alpha_{\mathcal{B}}^v \leq q_{\mathcal{B}|\emptyset} \vee v \text{ is } \mathcal{A}\text{-adopted as labeled in Eq.(2)} \end{cases}$$

Note that these diffusible labels are only based on a node's local state, and they are not limited to the nodes explored in the first phase — some nodes may only be explored in the second phase and they also need to be checked for these diffusible labels.

**Phase II: RR-set generation.** The second phase features a *primary backward search* from a random root $v$. Also, a number of secondary searches (from certain nodes explored in the primary search) may be necessary to find all nodes qualified for the RR-set. Intuitively, the primary backward search is to locate $\mathcal{A}$-suspended nodes $u$ via $\mathcal{AB}$-diffusible and $\mathcal{A}$-potential nodes, since once such a node $u$ adopts $\mathcal{B}$, it will adopt $\mathcal{A}$ and then through those $\mathcal{AB}$-diffusible and $\mathcal{A}$-potential nodes, the root $v$ will adopt $\mathcal{A}$ and $\mathcal{B}$. Thus such a node $u$ can be put into the RR-set of $v$. In addition, if such node $u$ is also $\mathcal{AB}$-diffusible, then any $\mathcal{B}$ seed $w$ that can activate $u$ to adopt $\mathcal{B}$ via $\mathcal{B}$-diffusible nodes can also be put into the RR-set of $v$, and we find such nodes $w$ using a secondary backward search from $u$ via $\mathcal{B}$-diffusible nodes. However, some additional complication may arise during the search process, and we cover all cases in Algorithm 3 and explain them below.

We first sample a root $v$ randomly from $V$. In case $v$ is labeled $\mathcal{A}$-adopted or $\mathcal{A}$-rejected, we simply return $R$ as $\emptyset$ because no $\mathcal{B}$-seed set can change $v$'s adoption status of $\mathcal{A}$ (lines 2 to 3). The

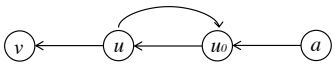



**Figure 3: Case 4 of primary backward search in RR-CIM**

primary search then starts. It first enqueues $v$ into a FIFO queue $Q$. Now consider a node $u$ dequeued from $Q$. Four cases arise.

*Case 1: $u$ is $\mathcal{A}$-suspended and $\mathcal{AB}$-diffusible* (lines 8 to 10). We add $u$ to $R$. Moreover, any node $w$ that can propagate $\mathcal{B}$ to $u$ by itself should also be added to $R$. To find all such $w$'s, we launch a secondary backward BFS from $u$ via $\mathcal{B}$-diffusible nodes. In particular, we conduct a reverse BFS from $u$ to explore all nodes that could reach $u$ via $\mathcal{B}$-diffusible nodes, and put all of them into set $R$. If this secondary search touches a node $w$ that is not $\mathcal{B}$-diffusible, we put $w$ in $R$ but do not further explore the in-neighbors of $w$.

*Case 2: $u$ is $\mathcal{A}$-suspended but not $\mathcal{AB}$-diffusible* (line 11). Add $u$ to $R$, but do not initiate a secondary search, because $u$ cannot adopt $\mathcal{A}$ or $\mathcal{B}$ even if it is informed of both $\mathcal{A}$ and $\mathcal{B}$, and thus the only way to make it adopt $\mathcal{B}$ is to make it a $\mathcal{B}$ seed.

*Case 3: $u$ is $\mathcal{A}$-potential and $\mathcal{AB}$-diffusible* (lines 13 to 15). We enqueue all $w \in N^-(u)$ such that $(w, u)$ is live in $W$ and continue without adding $u$ to $R$, as $u$ cannot even be informed of $\mathcal{A}$ by $S_\mathcal{A}$ alone, and hence cannot propagate $\mathcal{A}$ to $v$ by itself.

*Case 4: $u$ is $\mathcal{A}$-potential but not $\mathcal{AB}$-diffusible* (lines 16 to 19). This is the most complicated case that needs a special treatment. In general, we should stop the primary backward search at $u$ and try other branches, because $u$ is not yet $\mathcal{A}$-informed and $u$ cannot help in diffusing $\mathcal{A}$ and $\mathcal{B}$ even when informed of $\mathcal{A}$ and $\mathcal{B}$. However, there is a special case in which we can still put $u$ in $R$ (making $u$ a $\mathcal{B}$-seed): $u$ can reach an $\mathcal{A}$-suspended and $\mathcal{AB}$-diffusible node $u_0$ via a $\mathcal{B}$-diffusible path such that $u$ can activate $u_0$ in adopting $\mathcal{B}$ through this path, and then $u_0$ can reach back $u$ via an $\mathcal{AB}$-diffusible path, such that $u_0$ can activate $u$ in adopting $\mathcal{A}$. E.g., consider Figure 3 (all edges are live): $a$ is an $\mathcal{A}$-seed, $u$ is $\mathcal{A}$-potential but not $\mathcal{AB}$-diffusible, and $u_0$ is $\mathcal{AB}$-diffusible and $\mathcal{A}$-suspended.

To identify such $u$, we start two secondary BFS from $u$, one traveling forwards, one backwards. The forward search explores all $\mathcal{B}$-diffusible nodes reachable from $u$ and puts them in a set $S_f$, and stops at a node $w$ when $w$ is not $\mathcal{B}$-diffusible, but also puts $w$ in set $S_f$. The backward search explores all $\mathcal{AB}$-diffusible and $\mathcal{A}$-potential/suspended/adopted nodes that can reach $u$ and puts them in set $S_b$. If there is a node $u_0 \in S_f \cap S_b$ that is $\mathcal{A}$-suspended, then we can put $u$ into $R$. After this special treatment, we stop exploring the in-neighbors of $u$ in the primary search and continue the primary search elsewhere. We have:

THEOREM 8. *Suppose that $q_{\mathcal{A}|\emptyset} \leq q_{\mathcal{A}|\mathcal{B}}$ and $q_{\mathcal{B}|\emptyset} \leq q_{\mathcal{B}|\mathcal{A}} = 1$. The RR-sets generated by the* RR-CIM *algorithm satisfies Definition 1 for the* CompInfMax *problem. As a result, Theorem 6 applies to* GeneralTIM *with* RR-CIM *in this case.*

**Expected Time Complexity.** Both phases of RR-CIM require more computations compared to RR-SIM. First, the number of edges explored in Phase I, namely $EPT_F$, is larger in RR-CIM, as the forward labeling here needs to continue beyond just $\mathcal{A}$-adopted nodes. For Phase II, let $EPT_{BS}$ be the expected number of edges pointing to nodes in $R$ and $EPT_{BO}$ be the expected number of all other edges examined in this phase (including both primary and secondary searches). Thus, we have:

LEMMA 7. *In expectation,* GeneralTIM *with* RR-CIM *runs in $O\left((k+\ell)(|V|+|E|)\log|V|\left(1+\frac{EPT_F+EPT_{BO}}{EPT_{BS}}\right)\right)$ time.*

## 6.4 The Sandwich Approximation Strategy

We present the *Sandwich Approximation* (SA) strategy that leads to algorithms with data-dependent approximation factors for SelfInfMax and CompInfMax in the general mutual complement case of Com-IC ($q_{\mathcal{A}|\emptyset} \leq q_{\mathcal{A}|\mathcal{B}}$ and $q_{\mathcal{B}|\emptyset} \leq q_{\mathcal{B}|\mathcal{A}}$) when submodularity may *not* hold. In fact, SA can be seen as a general strategy, applicable to any non-submodular maximization problems for which we can find submodular upper or lower bound functions.

Let $\sigma : 2^V \to \mathbb{R}_{\geq 0}$ be non-submodular. Let $\mu$ and $\nu$ be submodular and defined on the same ground set $V$ such that $\mu(S) \leq \sigma(S) \leq \nu(S)$ for all $S \subseteq V$. That is, $\mu$ ($\nu$) is a lower (resp., upper) bound on $\sigma$ everywhere. Consider the problem of maximizing $\sigma$ subject to a cardinality constraint $k$. Notice that if the objective function were $\mu$ or $\nu$, the problem would be approximable within $1-1/e$ (e.g., max-$k$-cover) or $1-1/e-\epsilon$ (e.g., influence maximization) by the greedy algorithm [16, 21]. A natural question is: *Can we leverage the fact that $\mu$ and $\nu$ "sandwich" $\sigma$ to derive an approximation algorithm for maximizing $\sigma$?* The answer is "yes".

**Sandwich Approximation.** First, run the greedy algorithm on all three functions. It produces an approximate solution for $\mu$ and $\nu$. Let $S_\mu, S_\sigma, S_\nu$ be the solution obtained for $\mu, \sigma$, and $\nu$ respectively. Then, select the final solution to $\sigma$ to be

$$S_{sand} = \underset{S \in \{S_\mu, S_\sigma, S_\nu\}}{\arg\max} \sigma(S). \tag{3}$$

THEOREM 9. *Sandwich Approximation solution gives:*

$$\sigma(S_{sand}) \geq \max\left\{\frac{\sigma(S_\nu)}{\nu(S_\nu)}, \frac{\mu(S^*_\sigma)}{\sigma(S^*_\sigma)}\right\} \cdot (1-1/e) \cdot \sigma(S^*_\sigma), \tag{4}$$

*where $S^*_\sigma$ is the optimal solution maximizing $\sigma$ (subject to cardinality constraint $k$).*

**Remarks.** While the factor in Eq. (4) involves $S^*_\sigma$, generally not computable in polynomial time, the first term inside max{.,.}, involves $S_\mu$ can be computed efficiently and can be of practical value (see Table 4 in §7). We emphasize that SA is much more general, not restricted to cardinality constraints. E.g., for a general matroid constraint, simply replace $1-1/e$ with $1/2$ in (4), as the greedy algorithm is a $1/2$-approximation in this case [21]. Furthermore, monotonicity is not important, as maximizing general submodular functions can be approximated within a factor of $1/2$ [5], and thus SA applies regardless of monotonicity. On the other hand, the true effectiveness of SA depends on how close $\nu$ and $\mu$ are to $\sigma$: e.g., a constant function can be a trivial submodular upper bound function but would only yield trivial data-dependent approximation factors. Thus, an interesting question is how to derive $\nu$ and $\mu$ that are as close to $\sigma$ as possible while maintaining submodularity.

**SelfInfMax.** GeneralTIM with RR-SIM or RR-SIM+ provides a $(1-1/e-\epsilon)$-approximate solution with high probability, when $q_{\mathcal{A}|\emptyset} \leq q_{\mathcal{A}|\mathcal{B}}$ and $q_{\mathcal{B}|\emptyset} = q_{\mathcal{B}|\mathcal{A}}$. When $q_{\mathcal{B}|\emptyset} < q_{\mathcal{B}|\mathcal{A}}$, function $\nu$ (upper bound) can be obtained by increasing $q_{\mathcal{B}|\emptyset}$ to $q_{\mathcal{B}|\mathcal{A}}$, while $\mu$ (lower bound) can be obtained by decreasing $q_{\mathcal{B}|\mathcal{A}}$ to $q_{\mathcal{B}|\emptyset}$.

**CompInfMax.** GeneralTIM with RR-CIM provides a $(1-1/e-\epsilon)$-approximate solution with high probability, when $q_{\mathcal{A}|\emptyset} \leq q_{\mathcal{A}|\mathcal{B}}$ and $q_{\mathcal{B}|\emptyset} \leq q_{\mathcal{B}|\mathcal{A}} = 1$. When $q_{\mathcal{B}|\mathcal{A}}$ is not necessarily 1, we obtain an upper bound function by increasing $q_{\mathcal{B}|\mathcal{A}}$ to 1.

THEOREM 10. *Suppose $q_{\mathcal{A}|\emptyset} \leq q_{\mathcal{A}|\mathcal{B}}$ and $q_{\mathcal{B}|\emptyset} \leq q_{\mathcal{B}|\mathcal{A}}$. Then, under the Com-IC model, for any fixed $\mathcal{A}$ and $\mathcal{B}$ seed sets $S_\mathcal{A}$ and $S_\mathcal{B}$, $\sigma_\mathcal{A}(S_\mathcal{A}, S_\mathcal{B})$ is monotonically increasing w.r.t. any one of $\{q_{\mathcal{A}|\emptyset}, q_{\mathcal{A}|\mathcal{B}}, q_{\mathcal{B}|\emptyset}, q_{\mathcal{B}|\mathcal{A}}\}$ with other three GAPs fixed, as long as after the increase the parameters are still in $\mathbf{Q}^+$.*

| | Douban-Book | Douban-Movie | Flixster | Last.fm |
|---|---|---|---|---|
| # nodes | 23.3K | 34.9K | 12.9K | 61K |
| # edges | 141K | 274K | 192K | 584K |
| avg. out-degree | 6.5 | 7.9 | 14.8 | 9.6 |
| max. out-degree | 1690 | 545 | 189 | 1073 |

**Table 1: Statistics of graph data (all directed)**

Putting it all together, the final algorithm for SELFINFMAX is GeneralTIM with RR-SIM/RR-SIM+ and SA. Similarly, the final algorithm for COMPINFMAX is Algorithm 1 with RR-CIM and SA. It is important to see how useful and effective SA is in practice. We address this question head on in §7, where we "stress test" the idea behind SA. Intuitively, if the GAPs are such that $q_{\mathcal{B}|\emptyset}$ and $q_{\mathcal{B}|\mathcal{A}}$ are close, the upper and lower bounds ($\nu$ and $\mu$) obtained for SELFINFMAX can be expected to be quite close to $\sigma$. Similarly, when $q_{\mathcal{B}|\mathcal{A}}$ is close to 1, the corresponding upper bound for COMPINFMAX should be quite close to $\sigma$. We consider settings where $q_{\mathcal{B}|\emptyset}$ and $q_{\mathcal{B}|\mathcal{A}}$ are separated apart and similarly $q_{\mathcal{B}|\mathcal{A}}$ is not close to 1 and measure the effectiveness of SA (see Table 4).

# 7. EXPERIMENTS

We perform extensive experiments on three real-world social networks. We first present results with synthetic GAPs (§7.1); then we propose a method for learning GAPs using action log data (§7.2), and conduct experiments using learnt GAPs (§7.3).

**Datasets.** Flixster is collected from a social movie site and we extract a strongly connected component. Douban is collected from a Chinese social network [26], where users rate books, movies, music, etc. We crawl all movie & book ratings of the users in the graph, and derive two datasets from book and movie ratings: Douban-Book and Douban-Movie. Last.fm is taken from the popular music website with social networking features. Table 1 presents basic stats of the datasets. For all graphs, we learn influence probabilities on edges using the method proposed in [13], which is widely adopted in prior work [9]. Links in Flixster and Last.fm networks are undirected, and we direct them in both directions. Links in Douban network are derived from follower-followee relationships and in our dataset, there is an edge from $u$ to $v$ if $v$ follows $u$ on Douban.

## 7.1 Experiments with Synthetic GAPs

We first evaluate our proposed algorithms using synthetic GAPs. We compare with two intuitive baselines: $(i)$ VanillaIC: It selects $k$ seeds using TIM algorithm [25] under the classic IC model, essentially ignoring the other product and the NLA in Com-IC model; $(ii)$ Copying: For SELFINFMAX, it simply selects the top-$k$ $\mathcal{B}$-seeds to be $\mathcal{A}$-seeds and vice versa for COMPINFMAX.

In SELFINFMAX, we set $q_{\mathcal{A}|\mathcal{B}} = q_{\mathcal{B}|\mathcal{A}} = 0.75$, $q_{\mathcal{B}|\emptyset} = 0.5$, and $q_{\mathcal{A}|\emptyset}$ is set to 0.1, 0.3, 0.5, which represent strong, moderate, and low complementarity. In COMPINFMAX, we set $q_{\mathcal{A}|\emptyset} = 0.1$, $q_{\mathcal{A}|\mathcal{B}} = q_{\mathcal{B}|\mathcal{A}} = 0.9$, such that the room for $\mathcal{B}$ to complement $\mathcal{A}$ is sufficiently large to distinguish between algorithms. We vary $q_{\mathcal{B}|\emptyset}$ to be 0.1, 0.5, and 0.8.

Lots of possibilities exist for setting the opposite seed set, i.e., $\mathcal{B}$-seeds for SELFINFMAX and $\mathcal{A}$-seeds for COMPINFMAX. We test three representative cases: (1) randomly selecting 100 nodes – this models our complete lack of knowledge; (2) running VanillaIC and selecting the top-100 nodes – this models a situation where we assume the advertiser might use an advanced algorithm such as TIM to target highly influential users; (3) running VanillaIC and selecting the 101st to 200th nodes – this models a situation where we assume those seeds are moderately influential.

Table 2 shows the percentage improvement of our algorithms over the two baselines, for the case of selecting the 101st to 200th nodes of VanillaIC as the fixed opposite seed set (two other cases are discussed in [1]). As can be seen, GeneralTIM performs consistently better than both baselines, and in many cases by a large margin. Furthermore, our algorithms are especially effective when the opposite fixed seed set ($\mathcal{B}$-seeds for SELFINFMAX and $\mathcal{A}$-seeds for COMPINFMAX) are not optimally selected. This is especially true in real-world scenarios where the other product is released first and there are spontaneous early adopters.

Also, in our model the influence probabilities on edges are assumed independent of the product; without this assumption Copying and VanillaIC would perform even more poorly. If we additionally assume that the GAPs are user-dependent, VanillaIC would deteriorate further. In contrast, our GeneralTIM and RR-set generation algorithms can be easily adapted to both these scenarios.

| $\mathcal{A}$ | $\mathcal{B}$ | $q_{\mathcal{A}\|\emptyset}$ | $q_{\mathcal{A}\|\mathcal{B}}$ | $q_{\mathcal{B}\|\emptyset}$ | $q_{\mathcal{B}\|\mathcal{A}}$ |
|---|---|---|---|---|---|
| *Monster Inc.* | *Shrek* | .88 | .92 | .92 | .96 |
| *Gone in 60 Seconds* | *Armageddon* | .63 | .77 | .67 | .82 |
| *Prisoner of Azkaban* | *What a Girl Wants* | .85 | .84 | .66 | .67 |
| *Shrek* | *Fast and Furious* | .92 | .94 | .80 | .79 |

**Table 3: Selected GAPs learned for movies from Flixster**

## 7.2 Learning GAPs from Real Data

**Finding Signals from Data.** For Flixster and Douban, we learn GAPs from timestamped rating data, which can be viewed as action logs. Each entry is a quadruple $(u, i, a, t_{u,i,a})$, indicating user $u$ performed action $a$ on item $i$ at time $t_{u,i,a}$. We count a rating quadruple as one *adoption action* and one *informing action*: if someone rated an item, she must have been informed of it first, as we assume only adopters rate items. A key challenge is how to find actions that can be mapped to informing events that do not lead to adoptions. Fortunately, there are special ratings providing such signals in Flixster and Douban. The former allows users to indicate if they "want to see" a movie, or are "not interested" in one. We map both signals to the actions of a user being informed of a movie. The latter allows users to put items into a wish list. Thus, if a book/movie is in a user's wish list, we treat it as an *informing action*. For Douban, we separate actions on books and movies to derive two datasets: Douban-Book and Douban-Movie.

**Learning Method.** Consider two items $\mathcal{A}$ and $\mathcal{B}$ in an action log. Let $R_{\mathcal{A}}$ and $I_{\mathcal{A}}$ be the set of users who rated $\mathcal{A}$ and who were informed of $\mathcal{A}$, respectively. Clearly, $R_{\mathcal{A}} \subseteq I_{\mathcal{A}}$. Thus, $q_{\mathcal{A}|\emptyset} = |R_{\mathcal{A}} \setminus R_{\mathcal{B}\prec_{rate}\mathcal{A}}| \;/\; |I_{\mathcal{A}} \setminus R_{\mathcal{B}\prec_{inform}\mathcal{A}}|$, where $R_{\mathcal{B}\prec_{rate}\mathcal{A}}$ is the set of users who rated both items with $\mathcal{B}$ rated first, and $R_{\mathcal{B}\prec_{inform}\mathcal{A}}$ is the set of users who rated $\mathcal{B}$ before being informed of $\mathcal{A}$. Next, $q_{\mathcal{A}|\mathcal{B}}$ is computed as follows:$q_{\mathcal{A}|\mathcal{B}} = |R_{\mathcal{B}\prec_{rate}\mathcal{A}}| \;/\; |R_{\mathcal{B}\prec_{inform}\mathcal{A}}|$. Similarly, $q_{\mathcal{B}|\emptyset}$ and $q_{\mathcal{B}|\mathcal{A}}$ can be computed in a symmetric way.

Table 3 presents the GAPs learned for a few pairs of popular movies in Flixster dataset. More examples, including those in Douban-Book and Douban-Movie and 95%-confidence intervals of those estimates, can be found in [1].

## 7.3 Experiments with Learned GAPs

**Baselines.** We compare GeneralTIM with several baselines commonly used in the literature: HighDegree: choose the $k$ highest out-degree nodes as seeds; PageRank: choose the $k$ nodes with highest PageRank score; Random: choose $k$ seeds uniformly at random. We also include the Greedy algorithm [16] with 10K iterations of MC simulations to compute influence spread for the Com-IC diffusion processes. VanillaIC and Copying are omitted as the results are similar to those in §7.1 (when the GAPs are close to each other).

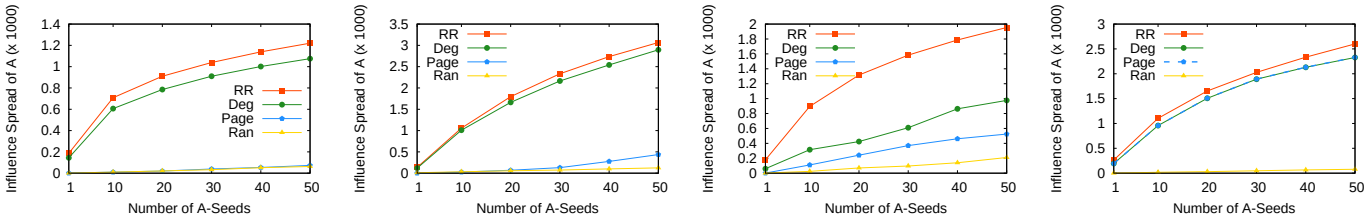
**Parameters.** The following pairs of items are tested:

- Douban-Book: *The Unbearable Lightness of Being* as $\mathcal{A}$ and *Norwegian Wood* as $\mathcal{B}$, and $\mathbf{Q} = \{0.75, 0.85, 0.92, 0.97\}$.
- Douban-Movie: *Fight Club* as $\mathcal{A}$ and *Se7en* as $\mathcal{B}$, and $\mathbf{Q} =$

| SELFINFMAX | VanillaIC | | | Copying | | |
|---|---|---|---|---|---|---|
| $q_{\mathcal{A}\|\emptyset}$ | 0.1 | 0.3 | 0.5 | 0.1 | 0.3 | 0.5 |
| Douban-Book | 5.89% | 0.93% | 0.50% | 85.7% | 207% | 301% |
| Douban-Movie | 24.7% | 3.30% | 1.72% | 13.3% | 68.8% | 122% |
| Flixster | 35.5% | 11.3% | 5.15% | 16.7% | 48.0% | 84.8% |
| Last.fm | 31.5% | 2.75% | 0.70% | 22.6% | 88.5% | 168% |

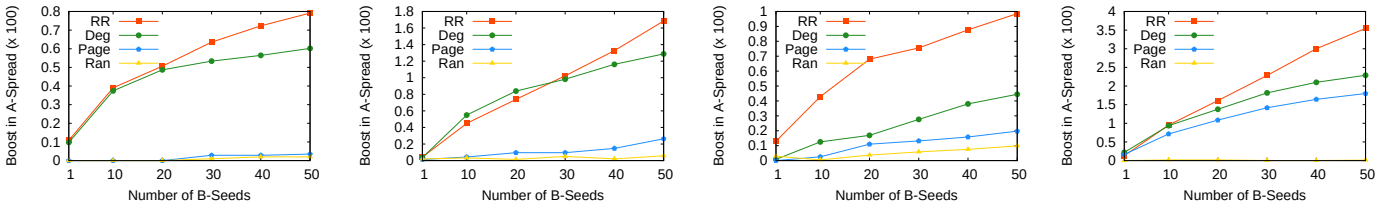
| COMPINFMAX | VanillaIC | | | Copying | | |
|---|---|---|---|---|---|---|
| $q_{\mathcal{B}\|\emptyset}$ | 0.1 | 0.5 | 0.8 | 0.1 | 0.5 | 0.8 |
| Douban-Book | 13.4% | 31.2% | 25.6% | 49.9% | 32.9% | 30.7% |
| Douban-Movie | 135% | 151% | 101% | 14.4% | 7.46% | 3.84% |
| Flixster | 81.7% | 58.5% | 24.9% | 13.6% | 8.21% | 10.7% |
| Last.fm | 140% | 110% | 48.3% | 10.6% | 9.12% | 6.77% |

**Table 2: Percentage improvement of GeneralTIM over VanillaIC & Copying**




(a) Douban-Book (b) Douban-Movie (c) Flixster (d) Last.fm

**Figure 4: $\mathcal{A}$-Spread vs. $|S_{\mathcal{A}}|$ for SELFINFMAX (RR – GeneralTIM with RR-SIM+, Page – PageRank; Deg – High-Degree, Rand – Random)**




(a) Douban-Book, $\sigma_{\mathcal{A}}(S_{\mathcal{A}}, \emptyset) = 612$ (b) Douban-Movie, $\sigma_{\mathcal{A}}(S_{\mathcal{A}}, \emptyset) = 2357$ (c) Flixster, $\sigma_{\mathcal{A}}(S_{\mathcal{A}}, \emptyset) = 1643$ (d) Last.fm, $\sigma_{\mathcal{A}}(S_{\mathcal{A}}, \emptyset) = 1568$

**Figure 5: Boost in $\mathcal{A}$-Spread vs. $|S_{\mathcal{B}}|$ for COMPINFMAX (RR – GeneralTIM with RR-CIM)**

$\{0.84, 0.89, 0.89, 0.95\}$.

• Flixster: *Monster Inc* as $\mathcal{A}$ and *Shrek* as $\mathcal{B}$.

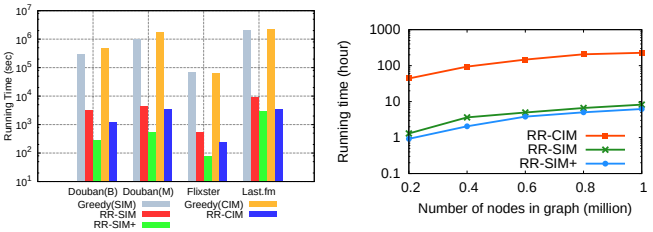
• Last.fm: There is no signal in the data to indicate informing events, so the learning method in §7.2 is not applicable. As a result, we use synthetic $\mathbf{Q} = \{0.5, 0.75, 0.5, 0.75\}$.

In all four datasets, $\mathcal{A}$ and $\mathcal{B}$ are mutually complementary, for which self/cross-submodularity does not hold (§5). Hence, Sandwich Approximation (SA) are used by default for GeneralTIM and Greedy [16]. Unless otherwise stated, $k = 50$. For GeneralTIM, $\ell = 1$ so that a success probability of $1 - 1/|V|$ is ensured [25]. In SELFINFMAX (resp. COMPINFMAX), the input $\mathcal{B}$-seeds (resp. $\mathcal{A}$-seeds) are chosen to be the 101st to 200th seeds selected by VanillaIC. We set $\epsilon = 0.5$, which is chosen to achieve a balance between efficiency (running time) and effectiveness (seed set quality). We empirically validate that influence spread is *almost completely unaffected* when $\epsilon$ varies from 0.1 to 1 [1].

Algorithms are implemented in C++ and compiled using g++ O3 optimization. We run experiments on an openSUSE Linux server with 2.93GHz CPUs and 128GB RAM.

**Quality of Seeds.** The quality of seeds is measured by the influence spread or boost achieved. We evaluate the spread of seed sets computed by all algorithms by MC simulations with 10K iterations for fair comparison. As can be seen from Figures 4 and 5, our RR-set algorithms are consistently the best in almost all test cases, often leading by a significant margin. The results of Greedy are omitted, since the spread it achieves is almost identical to GeneralTIM, matching the observations in prior work [25]. RR-SIM results are identical to RR-SIM+, and thus also omitted.

For SELFINFMAX, GeneralTIM with RR-SIM+ is 13%, 2.7%, 100%, and 13% better than the next best algorithm on Douban-Book, Douban-Movie, Flixster and Last.fm respectively, while for COMPINFMAX, GeneralTIM with RR-CIM is 31%, 31%, 122%, and 51% better. The boost in $\mathcal{A}$-spread provided by $\mathcal{B}$-seeds (GeneralTIM with RR-CIM) is at least 6% to 15% of the original $\mathcal{A}$-spread by $S_{\mathcal{A}}$ only. HighDegree performs well, especially in graphs with many nodes having large out-degrees (Douban-Movie, Last.fm), while PageRank produces good quality seeds only on Last.fm. Random is consistently the worst. The performances of baselines are generally consistent with observations in prior works [10, 11, 25] albeit for different diffusion models.




(a) Real networks (b) Synthetic graphs

**Figure 6: Running time**

**Running Time and Scalability.** We compare the running time of GeneralTIM to Greedy, shown in Figure 6(a). As can be seen, for SELFINFMAX, GeneralTIM with RR-SIM, RR-SIM+ is about *two to three orders of magnitude* faster than Greedy; for COMPINFMAX, GeneralTIM with RR-CIM is also about *two orders of magnitude* faster than Greedy. In addition, we observe that RR-SIM+ is 12, 8, 7, and 2 times as fast as RR-SIM on Douban-Book, Douban-Movie, Flixster, and Last.fm respectively. The running time of HighDegree, PageRank, and Random baselines are omitted since they are typically very efficient [9–11].

We then use larger synthetic networks to test the scalability of GeneralTIM with our RR-set generation algorithms. We generate power-law random graphs of 0.2, 0.4, ..., up to 1 million nodes with a power-law degree exponent of 2.16 [10]. These graphs have an average degree of about 5. We use the GPAs from Flixster. We can see that GeneralTIM with RR-SIM+ within 6.2 hours for the 1-million node graph, and its running time grows linearly in graph

| | Douban-Book | Douban-Movie | Flixster | Last.fm |
|---|---|---|---|---|
| $\mathsf{SIM}_{\mathsf{learn}}$ | 0.996 | 0.999 | 0.996 | 0.999 |
| $\mathsf{SIM}_{0.1}$ | 0.652 | 0.962 | 0.492 | 0.519 |
| $\mathsf{SIM}_{0.5}$ | 0.770 | 0.969 | 0.633 | 0.628 |
| $\mathsf{SIM}_{0.9}$ | 0.946 | 0.985 | 0.926 | 0.879 |
| $\mathsf{CIM}_{\mathsf{learn}}$ | 0.973 | 0.918 | 0.950 | 0.825 |
| $\mathsf{CIM}_{0.1}$ | 0.913 | 0.832 | 0.933 | 0.772 |
| $\mathsf{CIM}_{0.5}$ | 0.936 | 0.885 | 0.969 | 0.857 |
| $\mathsf{CIM}_{0.9}$ | 0.956 | 0.976 | 0.993 | 0.959 |

**Table 4: Sandwich approximation:** $\sigma(S_\nu)/\nu(S_\nu)$

size, which indicates great scalability. RR-CIM is slower due to the inherent intricacy of CompInfMax, but it also scales linearly. To put its running time measures in perspective, Greedy— the only other known approximation algorithm for CompInfMax— takes about 48 hours on Flixster (12.9K nodes), while GeneralTIM with RR-CIM is 4 hours faster on a graph 10 times as large.

**Approximation Factors by Sandwich Approximation.** Recall from §6.4 that the approximation factor yielded by SA is data-dependent: $\sigma(S_{sand}) \geq \max\{\frac{\sigma(S_\nu)}{\nu(S_\nu)}, \frac{\mu(S^*_\sigma)}{\sigma(S^*_\sigma)}\}\cdot(1-1/e-\epsilon)\cdot\sigma(S^*_\sigma)$. To see how good the SA approximation factor is in real-world graphs, we compute $\sigma(S_\nu)/\nu(S_\nu)$, as SA is guaranteed to have an approximation factor of at least $(1-1/e-\epsilon)\cdot\sigma(S_\nu)/\nu(S_\nu)$.

In the GAPs learned from data, both $q_{\mathcal{B}|\mathcal{A}}-q_{\mathcal{B}|\emptyset}$ and $q_{\mathcal{A}|\mathcal{B}}-q_{\mathcal{A}|\emptyset}$ are small and thus likely "friendly" to SA, as we mentioned in §6.4. Thus, we further "stress test" SA with more adversarial settings: First, set $q_{\mathcal{A}|\emptyset}=0.3$ and $q_{\mathcal{A}|\mathcal{B}}=0.8$; Then, for SelfInfMax, fix $q_{\mathcal{B}|\mathcal{A}}=1$ and vary $q_{\mathcal{B}|\emptyset}$ from $\{0.1, 0.5, 0.9\}$; for CompInfMax, fix $q_{\mathcal{B}|\emptyset}=0.1$ and vary $q_{\mathcal{B}|\mathcal{A}}$ from $\{0.1, 0.5, 0.9\}$.

Table 4 illustrates the results on all datasets with both learned GAPs and artificial GAPs. We use shorthands SIM and CIM for SelfInfMax and CompInfMax respectively. Subscript learn means the GAPs are learned from data. In stress-test cases (other six rows), e.g., for SIM, subscript 0.5 means $q_{\mathcal{B}|\emptyset}=.5$, while for CIM, it means $q_{\mathcal{B}|\mathcal{A}}=.5$. As can be seen, with real GAPs, the ratio is extremely close to 1, matching our intuition. For artificial GAPs, the ratio is not as high, but most of them are still close to 1. E.g., in the case of $\mathsf{SIM}_{0.5}$, $\sigma(S_\nu)/\nu(S_\nu)$ ranges from 0.628 (Last.fm) to 0.969 (Douban-Movie), which correspond to an approximation factor of 0.40 and 0.61 ($\epsilon$ omitted). Even the smallest ratio (0.492 in $\mathsf{SIM}_{0.1}$, Flixster) would still yield a decent factor at about 0.3. This shows that SA is fairly effective and robust for solving non-submodular cases of SelfInfMax and CompInfMax. In [1], we provide further evidences to support our findings here.

# 8. CONCLUSIONS & FUTURE WORK

In this work, we propose the Comparative Independent Cascade (Com-IC) model that allows any degree of competition or complementarity between two different propagating items, and study the novel SelfInfMax and CompInfMax problems for complementary products. We develop non-trivial extensions to the RR-set techniques to achieve approximation algorithms. For non-submodular settings, we propose Sandwich Approximation to achieve data-dependent approximation factors. Our experiments demonstrate the effectiveness and efficiency of proposed solutions.

For future work, one direction is to design more efficient algorithms or heuristics for SelfInfMax and (especially) CompInfMax; e.g., whether near-linear time algorithm is still available for these problems is still open. Another direction is to fully characterize the entire GAP space $\mathbf{Q}$ in terms of monotonicity and submodularity properties. Moreover, an important direction is to extend the model to multiple items. Given the current framework, Com-IC can be extended to accommodate $k$ items, if we allow $k\cdot 2^{k-1}$ GAP parameters — for each item, we specify the probability of adoption for every combination of other items that have been adopted. However, how to simplify the model and make it tractable, how to reason about the complicated two-way or multi-way competition and complementarity, how to analyze monotonicity and submodularity, and how to learn GAP parameters from real-world data all remain as interesting challenges. Last, it is also interesting to consider an extended Com-IC model in which influence probabilities on edges are product-dependent.

**Acknowledgments.** This research is supported in part by a Discovery grant and a Discovery Accelerator Supplements grant from the Natural Sciences and Engineering Research Council of Canada (NSERC). We also thank Lewis Tzeng for some early discussions on modeling influence propagations for partially competing and partially complementary items.

# 9. REFERENCES

[1] http://arxiv.org/abs/1507.00317.
[2] S. Bharathi, D. Kempe, M. Salek. Competitive influence maximization in social networks. In *WINE*, 2007.
[3] C. Borgs et al. Maximizing social influence in nearly optimal time. In *SODA*, pages 946–957, 2014.
[4] A. Borodin et al. Threshold models for competitive influence in social networks. In *WINE*, pages 539–550, 2010.
[5] N. Buchbinder et al. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *FOCS*, 2012.
[6] C. Budak, D. Agrawal, A. El Abbadi. Limiting the spread of misinformation in social networks. In *WWW*, 2011.
[7] T. Carnes et al. Maximizing influence in a competitive social network: a follower's perspective. In *ICEC*, pages 351–360, 2007.
[8] W. Chen, et al. Influence maximization in social networks when negative opinions may emerge and propagate. In *SDM*, 2011.
[9] W. Chen, L. V. S. Lakshmanan, and C. Castillo. *Information and Influence Propagation in Social Networks*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2013.
[10] W. Chen, et al. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, 2010.
[11] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, 2010.
[12] S. Datta, A. Majumder, and N. Shrivastava. Viral marketing for multiple products. In *ICDM 2010*, pages 118–127, 2010.
[13] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, pages 241–250, 2010.
[14] X. He et al., Influence blocking maximization in social networks under the competitive linear threshold model. In *SDM*, 2012.
[15] S. Kalish. A new product adoption model with price, advertising, and uncertainty. *Management Science*, 31(12):1569–1585, 1985.
[16] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.
[17] W. Lu, et al. The bang for the buck: fair competitive viral marketing from the host perspective. In *KDD*, 2013.
[18] J. J. McAuley, R. Pandey, and J. Leskovec. Inferring networks of substitutable and complementary products. In *KDD*, 2015.
[19] S. A. Myers and J. Leskovec. Clash of the contagions: Cooperation and competition in information diffusion. In *ICDM*, 2012.
[20] R. Narayanam and A. A. Nanavati. Viral marketing for product cross-sell through social networks. In *PKDD*, 2012.
[21] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.
[22] N. Pathak, A. Banerjee, and J. Srivastava. A generalized linear threshold model for multiple cascades. In *ICDM*, 2010.
[23] C. Snyder and W. Nicholson. *Microeconomic Theory, Basic Principles and Extensions (10th ed)*. 2008.
[24] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: a martingale approach. In *SIGMOD*, pages 1539–1554, 2015.
[25] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: near-optimal time complexity meets practical efficiency. In *SIGMOD*, 2014.
[26] N. Yuan et al. We know how you live: exploring the spectrum of urban lifestyles. In *COSN*, 2013.