Proceedings of the Eighteenth International AAAI Conference on Web and Social Media (ICWSM 2024)

Cascade-based Randomization for Inferring Causal Effects under Diffusion Interference

Zahra Fatemi[1], **Jean Pouget-Abadie**[2], **Elena Zheleva**[1]
[1]University of Illinois Chicago
[2]Google Research
zfatem2@uic.edu, jeanpa@google.com, ezheleva@uic.edu

Abstract

The presence of interference, where the outcome of an individual may depend on the treatment assignment and behavior of neighboring nodes, can lead to biased causal effect estimation. Current approaches to network experiment design focus on limiting interference through cluster-based randomization, in which clusters are identified using graph clustering, and cluster randomization dictates the node assignment to treatment and control. However, cluster-based randomization approaches perform poorly when interference propagates in cascades, whereby the response of individuals to treatment propagates to their multi-hop neighbors. When we have knowledge of the cascade seed nodes, we can leverage this interference structure to mitigate the resulting causal effect estimation bias. With this goal, we propose a cascade-based network experiment design that initiates treatment assignment from the cascade seed node and propagates the assignment to their multi-hop neighbors to limit interference during cascade growth and thereby reduce the overall causal effect estimation error. Our extensive experiments on real-world and synthetic datasets demonstrate that our proposed framework outperforms the existing state-of-the-art approaches in estimating causal effects in network data.

Introduction

Randomized Controlled Trials (RCTs), also known as A/B tests, are considered the gold standard for inferring causality (Antman et al. 1992; Aral and Walker 2011; Bakshy et al. 2012; Ho 2017; Betts et al. 2014; Kazdin 2022). Through randomization, the experimenter can ensure that treatment and control assignments are independent of other variables. As a motivating example, consider a company that aims to assess the effectiveness of promoting its products through viral marketing (also known as word-of-mouth marketing) using an RCT. The campaign under consideration offers customers who have purchased a new phone (to which we refer as "influential nodes") a bill discount in exchange for sharing positive feedback about the product on their social media platforms. To infer the causal effect of the campaign on product adoption, influential nodes are randomly split into a treatment group and a control group. The treatment group is exposed to the treatment (i.e., discount offer) and the control group is not. Then the product adoption rate in the social network of the treatment group nodes is compared to that in the social network of the control group nodes.

One of the challenges with designing a randomized controlled trial like this is that it is hard to isolate parts of a social network for treatment and control without any interactions. This is especially true when considering that word of mouth can form multi-hop spread over the network. The presence of interference, where the outcome of a node in the control group can be influenced by the treatment nodes and vice versa, breaks a common causal inference assumption, known as the Stable Unit Treatment Value Assumption (SUTVA) (Rubin 1974; Imbens and Rubin 2015; Kohavi and Longbotham 2017). To address this problem, network experiment design focuses on ensuring reliable causal effect estimation in RCTs for potentially interacting units. Such network experiments have a wide range of applications across various disciplines, such as marketing (Aral and Walker 2011; Eckles, Karrer, and Ugander 2016; Holtz et al. 2020), and the medical (Halloran and Hudgens 2012; Shakya et al. 2017) and social sciences (Zhang et al. 2016; Glanz et al. 2017).

Cluster-based randomization or two-stage randomization approaches are prominent network experiment designs for minimizing interference (also known as spillover or network effects). They find densely connected clusters of nodes and assign treatment and control at the cluster level (Ugander et al. 2013; Eckles, Karrer, and Ugander 2017; Saveski et al. 2017; Pouget-Abadie et al. 2018; Ugander and Yin 2020; Fatemi and Zheleva 2020a). Despite their advantage in mitigating interference, these approaches assume 1-hop spillover (i.e., interference between immediate neighbors). However, a piece of information (e.g., about a new product or a meme) can propagate from nodes to their multi-hop neighbors over time through a cascade of influence in the network. Cascades are natural phenomena in social networks and are the consequence of network interactions. In the running example, early adopters can trigger a large word-of-mouth cascade in the network by influencing their friends on the social network to adopt the product, and their friends would influence their friends of friends, and so on. The presence of a cascade can intensify interference in network experiments because each individual gets exposed to the treatment and outcomes of not only its immediate neighbors but also its multi-hop neighbors over time. Various aspects of

network cascades have been studied in the research literature, from cascade growth prediction (Cheng et al. 2014; Li et al. 2017) to underlying network prediction (Netrapalli and Sanghavi 2012; Daneshmand et al. 2014; Pouget-Abadie and Horel 2015), and influence maximization (Chen, Wang, and Yang 2009; Wang, Chen, and Wang 2012; Chen et al. 2013; Zhao, Li, and Jin 2016). However, unbiased estimation of causal effects in cascade models remains an open problem.

Present Work. Motivated by applications in viral marketing and information diffusion, we propose a network experiment design to mitigate interference in cascade models. We specifically focus on the Independent Cascade Model (Kempe, Kleinberg, and Tardos 2003), a stochastic diffusion model where the adoption of a new product by a neighbor leads to neighboring nodes adopting the product with an independent probability. Under the assumption that cascade seed nodes (e.g., influential nodes) are known, we develop ***Cas**cade-**B**ased **R**andomization (CasBR)*, a network experiment design which assigns treatment starting from the cascade seed nodes and propagates the assignment to their multi-hop neighbors. The intuition is that by assigning cascade seed nodes and their multi-hop neighbors to the same treatment group, we can limit interference between treatment and control groups. Prior research has primarily utilized linear interference models and emphasized clustering as the optimal design for mitigating interference (Eckles, Karrer, and Ugander 2017; Saveski et al. 2017; Brennan, Mirrokni, and Pouget-Abadie 2022). In contrast, our paper challenges this approach by investigating a new interference model, the linear cascade model, and demonstrating that clustering may not be the most effective solution in the presence of a multi-hop cascade of influence. To summarize, this paper makes the following main contributions:

- We formulate the problem of causal effect estimation in the context of the Independent Cascade Model, specifically in scenarios where we are aware of the cascade seed nodes (i.e., initially active nodes) before conducting a controlled experiment.
- We empirically show that existing cluster-based randomization approaches lead to highly biased causal effect estimates when cascades of influence occur in the network.
- We propose a cascade-based framework, CasBR, that leverages cascade seed nodes to create clusters of seeds and their multi-hop neighbors, with the aim of limiting interference in network experiments on cascade models.

The rest of the paper is structured as follows. In Section 2, we review the background of causal effect estimation in network experiments. In Section 3, we define our data model, causal estimand of interest, and the problem that we address in the paper. In Section 4, we present our cascade-based network experiment design framework. In Section 5, we present the experimental setup and results in real-world and synthetic datasets. In Section 6, we conclude and discuss directions for future work.

Related Work

While it is straightforward to randomly assign nodes to treatment and control groups in i.i.d data, causal effect estimation in interacting units is challenging due to information spillover from treated nodes to control nodes (Eckles, Kizilcec, and Bakshy 2016; Aral and Walker 2011; Bailey et al. 2019; Shakya et al. 2017). A growing body of research has focused on minimizing the effects of spillover in network experiments including multilevel designs where treatment is applied with different proportions across the population (Hudgens and Halloran 2008; Tchetgen and VanderWeele 2012), model-based approaches under specific models of interference (Basse and Airoldi 2015; Aronow and Samii 2017; Toulis and Kao 2013; Choi 2017), methodologies that exclude neighboring nodes from the experiment (Fatemi and Zheleva 2020b, 2023), and experimental designs in two-sided platforms (Fradkin, Grewal, and Holtz 2021; Johari et al. 2022).

Several methods rely on graph clustering to deal with network interference (Eckles, Karrer, and Ugander 2017; Pouget-Abadie et al. 2019; Backstrom and Kleinberg 2011; Aronow and Middleton 2013; Ugander and Yin 2023; Candogan, Chen, and Niazadeh 2021). Cluster-based randomization approaches partition the graph into clusters and allocate treatment cluster by cluster. Ugander et al. propose a methodology using graph clustering to deal with interference and reduce the variance of the causal effect estimators (Ugander et al. 2013). Saveski et al. develop a cluster-based model with balanced clusters to test for the presence of network interference in large-scale experimentation platforms (Saveski et al. 2017). Saint-Jacques et al. develop an ego-network randomization approach where a cluster is comprised of an ego and its immediate neighbors to measure the network effects and relax strong modeling assumptions existing in prior works (Saint-Jacques et al. 2019). Recent studies show the advantage of cluster-based randomization approach in causal effect estimation in bipartite graphs under interference (Brennan, Mirrokni, and Pouget-Abadie 2022; Harshaw et al. 2021)

A new line of research shows that there is a trade-off between interference and selection bias in cluster-based randomization approaches based on the chosen number of clusters (Fatemi and Zheleva 2020a, 2023). This work develops a network experiment design framework that combines node matching with weighted graph clustering to jointly minimize interference and selection bias. Since our work is closest to cluster-based randomization approaches, we use these methods as baselines in our experiments. Inspired by Kempe, Kleinberg, and Tardos (2003), there are numerous studies on cascades from an influence maximization perspective (Chen, Wang, and Yang 2009; Morone and Makse 2015; Bharathi, Kempe, and Salek 2007). This line of work focuses more on influence maximization than the estimation of causal effects when influence propagates. On the topic of estimation, there is another line of work that seeks to estimate the graphs along which cascade models propagate (Gomez-Rodriguez, Leskovec, and Krause 2012). However, in our paper, we assume the graph is known and we seek to estimate the impact of cascades on network experiments. Recently, Poiitis,

Vakali, and Kourtellis (2021) have looked into the value of choosing different seed node set sizes on aggression diffusion modeling and found that different seed sizes only affect the duration of the diffusion process but not the accuracy of user aggression scores.

Problem Setup

In this section, we formally define the data model, the causal estimand, and the problem we address in this paper.

Data Model

We consider a graph $G = \{\mathbf{V}, \mathbf{E}\}$ with a set of n nodes $\mathbf{V}$ and a set of edges $\mathbf{E} = \{e_{ij}\}$ where e_{ij} denotes that there is an edge between node $v_i \in \mathbf{V}$ and node $v_j \in \mathbf{V}$. Each node $v_i \in \mathbf{V}$ has an associated binary outcome variable $Y_i \in 0, 1$ and a treatment assignment variable $T_i \in 0, 1, 2$, with $T_i = 1$ indicating that the node is in treatment, $T_i = 0$ indicating that the node is in control, and $T_i = 2$ indicating that the node is excluded from the experiment. If node v_i is activated (e.g., adopted the product), then $Y_i = 1$. We define $\mathbf{Z} \in \{0, 1, 2\}^n$ as the treatment assignment vector of all nodes. We further denote N_i as the set of neighbors of node v_i.

Independent Cascade Model

The *Independent Cascade Model (ICM)* is a stochastic information diffusion model where information spreads through a cascade across the nodes of a graph (Kempe, Kleinberg, and Tardos 2003). In this model, each node can be in one of three states: 1) inactive, representing nodes that have not yet been activated by any of their neighboring active nodes, 2) active, representing nodes that were activated in the previous time step, and can activate inactive nodes in the next step, and 3) activated, representing nodes that activated other nodes in previous time steps and can no longer activate any other nodes. In the ICM, a cascade begins with a set of active seed nodes denoted by S_0, and at each time step t, a set of newly active nodes X_{t-1}^{new} is determined based on the previous time step. Each newly activated node can attempt to activate its inactive neighboring nodes with a certain propagation/spillover probability $p(v_i, v_j)$. If a node has multiple active neighbors in X_{t-1}^{new}, their attempts to activate the node are sequenced in an arbitrary order. The diffusion process continues until no further activations are possible. This paper assumes that the initial set of active cascade seed nodes S_0 is known.

Our cascade model includes four distinct spillover probabilities: 1) p_{t-t}, which represents the probability of a treatment node activating another treatment node; 2) p_{c-t}, the probability of a control node activating a treatment node; 3) p_{c-c}, the probability of a control node activating another control node; and 4) p_{t-c}, the probability of a treatment node activating a control node.

We now describe the interference model in network experiments on cascade models. In the first time step of cascade propagation, cascade seed nodes are activated. In section 5, we elaborate on two distinct methods for selecting the cascade seed nodes. After the treatment assignment mechanism is applied to the graph nodes, the following scenarios can occur based on the treatment assignment of an active node and its neighbor:

1. If the node and its neighbor are in treatment, the neighbor gets activated with probability p_{t-t}.
2. If the node and its neighbor are in control, the neighbor gets activated with probability p_{c-c}.
3. If the node is in control and its neighbor is in treatment, the neighbor gets activated with probability p_{c-t}.
4. If the node is in treatment and its neighbor is in control, the node gets activated with probability p_{t-c}.

The cascade propagation proceeds until all the m-hop neighbors of the seed nodes have been explored.

Types of Causal Effects Under Interference

The causal estimand of interest in this paper is the Total Treatment Effects (TTE), defined as the difference between the outcomes of individuals in two alternative universes, one in which everyone receives treatment ($\mathbf{z_1} = \{1\}^n$) and another where no one does ($\mathbf{z_0} = \{0\}^n$) (Ugander et al. 2013; Fatemi and Zheleva 2020a):

$$TTE = \frac{1}{n} \sum_{v_i \in V} (Y_i(\mathbf{Z} = \mathbf{z_1}) - Y_i(\mathbf{Z} = \mathbf{z_0})). \quad (1)$$

The fundamental problem for causal inference is that, for any individual, we can observe only the outcome under treatment or control group but not in both. Therefore, the effect $\hat{\tau}$ of a treatment on an outcome is typically calculated by averaging outcomes over treatment and control groups via difference-in-means: $\hat{\tau} = \overline{Y_{V_1}} - \overline{Y_{V_0}}$ where Y_{V_1} and Y_{V_0} denote the outcomes of treatment and control nodes, respectively (Halloran and Struchiner 1995; Stuart 2010; Fatemi and Zheleva 2020a).

In real-world scenarios and in the presence of interference, the measured TTE is a combination of direct treatment effects and peer effects. *Direct Treatment Effects (DTE)* is defined as the difference between the average outcomes of treated and untreated individuals due to the treatment alone and is measured as:

$$DTE(V) = \mathop{\mathbb{E}}_{v_i \in V}[Y_i | T_i = 1] - \mathop{\mathbb{E}}_{v_i \in V}[Y_i | T_i = 0]. \quad (2)$$

Average *peer effects (PE)* is the average influence of peers' behavior on the unit's response to the treatment and is estimated as:

$$PE(V) = \mathop{\mathbb{E}}_{v_i \in V}[Y_i | T_i = \omega, N_i.\boldsymbol{\pi}] - \mathop{\mathbb{E}}_{v_i \in V}[Y_i | T_i = \omega, N_i = \emptyset],$$

where $N_i.\boldsymbol{\pi}$ denotes the vector of treatment assignments to node v_i's neighbors N_i and ω shows the treatment assignment of v_i. Peer effects are divided into two categories (Fatemi and Zheleva 2020a):

- *Allowable peer effects (APE)*: APE is defined as the peer effects between nodes within the same treatment group

(e.g., two peers received a voucher) and is measured as:

$$APE(V) = \mathop{\mathbb{E}}_{v_i \in V}[Y_i | T_i = \omega, N_i^{\omega}.\boldsymbol{\pi}] - \mathop{\mathbb{E}}_{v_i \in V}[Y_i | T_i = \omega, N_i^{\omega} = \emptyset], \quad (3)$$

where N_i^{ω} denotes the set of neighbors of v_i in the same treatment group as node v_i with treatment assignment set $N_i^{\omega}.\boldsymbol{\pi}$.

- *Unallowable peer effects (UPE)*: UPE is defined as the peer effects between neighbors with different treatment assignments (e.g., two friends one received a voucher and the other one did not) and is measured as:

$$UPE(V) = \mathop{\mathbb{E}}_{v_i \in V}[Y_i | T_i = \omega, N_i^{\hat{\omega}}.\boldsymbol{\pi}] - \mathop{\mathbb{E}}_{v_i \in V}[Y_i | T_i = \omega, N_i^{\hat{\omega}} = \emptyset]. \quad (4)$$

where $N_i^{\hat{\omega}}$ ($\hat{\omega} \neq \omega$) denotes the set of neighbors in a different treatment group with treatment assignment $N_i^{\hat{\omega}}.\boldsymbol{\pi}$.

In the presence of allowable and unallowable peer effects in the network experiment, the measured TTE is a combination of DTE, APE, and UPE:

$$TTE = DTE(\mathbf{V}) + APE(\mathbf{V_1}) - APE(\mathbf{V_0}) + UPE(\mathbf{V_1}) - UPE(\mathbf{V_0}). \quad (5)$$

Where $\mathbf{V}_1$ and $\mathbf{V}_0$ represent sets of treatment and control nodes, respectively. Since APE are a natural consequence of network interactions, the main focus of this paper is to design a network experiment to minimize interference ($UPE(\mathbf{V_1}) \approx 0$ and $UPE(\mathbf{V_0}) \approx 0$) between nodes in treatment and control groups in such a way that the measured TTE represents an unbiased estimation of the true underlying causal effects.

Problem Definition

In this paper, we are interested in setting up a network experiment design to limit interference between treatment and control groups in such a way that $TTE \approx DTE(\mathbf{V}) + APE(\mathbf{V_1}) - APE(\mathbf{V_0})$. More formally:

Problem 1 *Network experiment design for the Independent Cascade Model. Given a graph $G = (\boldsymbol{V}, \boldsymbol{E})$, a set of attributes $\boldsymbol{V.X}$ associated with the graph nodes, a set of propagation probabilities $\boldsymbol{E.P}$ associated with the graph edges, and a set of active cascade seed nodes S_0, we would like to find a treatment assignment vector $\mathbf{Z}$ of a population with two different subsets of nodes, the treatment nodes $\boldsymbol{V}_1 \in \boldsymbol{V}$, and the control nodes $\boldsymbol{V}_0 \in \boldsymbol{V}$ such that:*

a. $\mathbf{V}_0 \cap \mathbf{V}_1 = \emptyset$*;*
b. $|\mathbf{V}_0| + |\mathbf{V}_1|$ *is maximized;*
c. $P(v_i \in \mathbf{V_1}) \times P(v_i \in \mathbf{V_0}) > 0$;
d. Multi-hop interference ($UPE(\boldsymbol{V}_1) - UPE(\boldsymbol{V}_0)$) is minimized.

The objective of the first component is to identify a set of treatment and control nodes that do not overlap with each other. The second component is designed to include as many nodes as possible from the entire node set $\boldsymbol{V}$, in order to obtain enough assignments to estimate the desired causal effects. The third component, referred to as the probabilistic/positivity assumption, requires that all units in the population under study have a positive chance of receiving any of the possible treatments. The fourth component aims to reduce multi-hop spillover in the experiment. It is worth noting that for certain methods of inference, having a positive overlap between treatment groups might not be adequate, and increasing the number of possible assignments across different treatment levels may improve the accuracy of the results.

Cascade-based Network Experiment Design

In this section, we present our network experiment design framework that mitigates unallowable peer effects (Eq. 4) in cascade models and provides a more accurate estimation of the causal effects (Eq. 1).

Cascade-Based Randomization Framework

With the goal of minimizing interference in the Independent Cascade Model, we propose ***Cas**cade-**B**ased **R**andomization (CasBR)*, a network experiment design framework that uses the cascade seed nodes as the seeds of the treatment randomization procedure to reduce the interference during cascade propagation in network experiments. Given the initial seeds of a cascade, CasBR consists of two main steps:

1. **Initial randomization**: In this step, the set of cascade seed nodes $\mathbf{S_0}$ are assigned to the treatment or control groups at random. This randomization allows the framework to create $|\mathbf{S_0}|$ clusters of treatment and control nodes with cascade seeds as the seed of each cluster.
2. **Assignment propagation**: In this step, treatment assignment is propagated in a multi-hop manner in the network, to satisfy requirement (d) of Problem 1. First, the immediate neighbors of the seed nodes are assigned to treatment or control. Considering the fraction of treated and untreated neighbors, each unassigned node at distance m from a cascade seed node is either assigned to treatment with probability $p_t(v_i)$ measured as:

$$p_t(v_i) = \frac{|treated_{ngb}(v_i)|}{|treated_{ngb}(v_i)| + |control_{ngb}(v_i)|}, \quad (6)$$

or assigned to control with probability $p_c(v_i) = 1 - p_t(v_i)$. $treated_{ngb}(v_i)$ stands for the set of treated neighbors of v_i and $control_{ngb}(v_i)$ denotes the set of neighbors in control. In order to avoid allocating all the budget to one treatment group, we create two distinct sets of unassigned neighbors of treatment and control nodes and then alternate between two sets. This fulfills requirement (a) of Problem 1.

Once the immediate neighbors of the seed nodes have been examined, we proceed to repeat the randomization process on the unassigned 2-hop neighbors of the seed nodes. This process is then repeated until all the 1-hop, 2-hop, and m-hop neighbors of the seed nodes have been assigned to either the treatment or control group. Singletons (i.e., isolated nodes without any neighbors) are assigned randomly to either the treatment or control

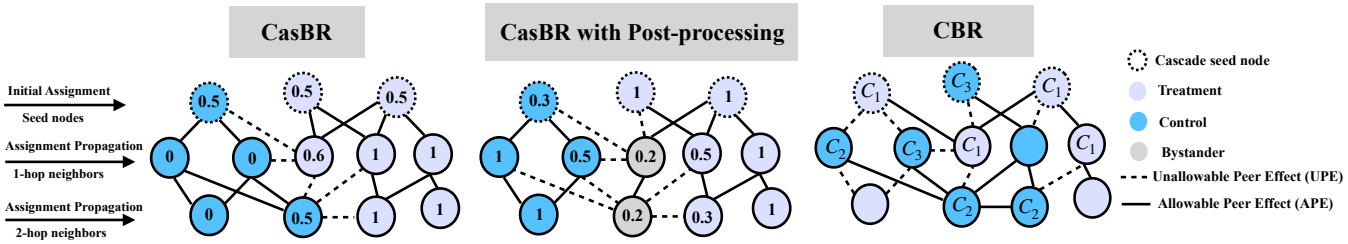


Figure 1: The resulting node assignment of three different network experiment designs for minimizing interference in the Independent Cascade Model. The CBR approach leaves more edges between treatment and control nodes compared to our CasBR method (9 vs. 5 edges) which indicates more interference in the experiment.

Algorithm 1: Cascade-based Randomization CasBR(G)

Input : $G(\mathbf{V}, \mathbf{E}), \mathbf{S_0}$
Output: $\mathbf{Z} = \{0,1\}^n$

1 Initialize $\mathbf{Z}$ by randomizing over cascade seeds$\mathbf{S_0}$;
2 **while** $\{\exists\ v_i \in \mathbf{V} : Z_j = \emptyset\}$ **do**
3 $\mathbf{N_t} = \{v_i | \exists\ e_{ji} \in \mathbf{E}\ \&\ Z_j = 1\ \&\ Z_i = \emptyset\}$;
4 $\mathbf{N_c} = \{v_i | \exists\ e_{ji} \in \mathbf{E}\ \&\ Z_j = 0\ \&\ Z_i = \emptyset\}$;
5 Shuffle($\mathbf{N_t}, \mathbf{N_c}$);
6 **while** $\{\mathbf{N_t} \neq \emptyset\ or\ \mathbf{N_c} \neq \emptyset\}$ **do**
7 Pop the first unassigned node v_i from $\mathbf{N_t}$;
8 $treated_{ngb} \leftarrow$ Set of immediate neighbors of v_i assigned to treatment;
9 $control_{ngb} \leftarrow$ Set of immediate neighbors of v_i assigned to control;
10 Calculate p_t based on $treated_{ngb}$ and $control_{ngb}$ using Eq.6;
11 Assign v_i to treatment with probability p_t;
12 $Z_i \leftarrow 1$ if v_i is assigned to treatment and $Z_i \leftarrow 0$ otherwise ;
13 Pop the first unassigned node v_i from $\mathbf{N_c}$;
14 Repeat Lines 8-12;
15 **return Z**

group. This satisfies requirement (b) of Problem 1. Since each seed node is randomly assigned to the treatment or control group, and any node, whether seed or non-seed, could have been assigned to either treatment group, our approach fulfills requirements (c) of Problem 1.

Post-processing: The objective of the CasBR framework is to allocate nodes and their multi-hop neighbors to a similar treatment group. However, in some cases, nodes may have multiple neighbors from both treatment and control groups, referred to as *bystander* nodes (e.g., gray nodes in Fig. 1). By excluding these nodes from the experiment, we may be able to reduce interference between treatment and control. However, peer effects from bystander nodes to treatment and control nodes may still occur via their peers. To mitigate this issue, we can choose bystander nodes with a more similar distribution of treatment and control neighbors. By doing so, the peer effects of bystander nodes on the treatment and control groups can cancel each other out in the total treatment effect estimation, leading to a reduction in the bias of the measured causal effects. To identify bystander nodes, we compute the difference between the fraction of treatment and control neighbors of each node v_i, denoted by β_i. If β_i is less than a threshold α, the node is added to the set of bystander nodes. Lower values of α indicate higher similarity between the distribution of treatment and control nodes in the neighborhood of bystander nodes. The measurement of β_i for each node v_i is defined as:

$$\beta_i = \frac{|treated_{ngb}(v_i)| - |control_{ngb}(v_i)|}{degree(v_i)} < \alpha. \quad (7)$$

Algorithm 2: Post-processing

Input : $G(\mathbf{V}, \mathbf{E}), \alpha$
Output: $\mathbf{Z} = \{0,1,2\}^n$

1 $\mathbf{Z} = \text{CasBR}(G)$
2 **for** $v_i \in \mathbf{V}$ **do**
3 **if** $\frac{|treated_{ngb}(v_i)| - |control_{ngb}(v_i)|}{degree(v_i)} < \alpha$ **then**
4 $Z_i \leftarrow 2$
5 **return Z**

Algorithm 1 and 2 depict the CasBR and post-processing algorithms as pseudo-code. The CasBR algorithm assigns nodes in $\mathbf{V}$ to either treatment or control. In Line 1 of Algorithm 1, the initial cascade seed nodes are randomly assigned to either treatment or control. In Lines 3-5, two sets of 1-hop (immediate) neighbors of treatment and control nodes are created and shuffled. In Lines 7-14, the probability of assigning each neighbor of seed nodes to treatment or control is calculated, and based on that, the node is assigned to treatment or control. The same process is then repeated for inactive 2-hop, ..., and m-hop neighbors of the cascade seeds to ensure that all nodes in the network are assigned to either treatment or control. In the worst-case scenario, m is equal to the diameter of the network which is the shortest distance between the two most distant nodes in the network. During the post-processing step (Algorithm 2), which is optional, a node is added to set $\mathbf{V}_2$ containing bystander nodes if the condition in Line 3 is met.

Fig. 1 presents a visualization of the node assignment for three different network experiment designs, using a toy example with three cascade seed nodes, five 1-hop neighbors,

and four 2-hop neighbors. In the first step of our CasBR approach, we randomly assign the cascade seed nodes (shown as dashed circles) to treatment or control with a probability $p_t = 0.5$. In the next step, we assign immediate unassigned neighbors of the seed nodes to either treatment or control with probability p_t calculated with Eq. 6. The numbers in each circle of the CasBR graph represent that probability of assigning each node to treatment at different distances from the cascade seed nodes. Similar to the 1-hop neighbors, we conduct randomization on the 2-hop neighbors of the seed nodes. Dashed lines show the presence of unallowable peer effects between endpoint nodes. The second experiment design is the same as the first one, except that we apply post-processing on CasBR and remove bystander nodes (shown as gray circles) from the experiment. The numbers in each circle of the post-processing graph are calculated using Eq. 7 where we set $\alpha = 0.3$. In the CBR approach, we partition the toy graph into three clusters using the reLDG clustering algorithm (Nishimura and Ugander 2013). Each circle in the CBR graph is annotated with the assigned cluster number. Nodes that remain unassigned by the reLDG algorithm are represented by circles without any annotation. These nodes are randomly assigned to the treatment or control group. The reason for poor clustering may be related to the fact that the algorithm is trying to find balanced clusters. This can lead to a situation where the algorithm prioritizes balancing the sizes of the clusters over finding densely connected clusters. As shown in Fig. 1, the CBR approach leads to a 57% increase (9 vs. 5 dashed edges) in the number of edges between treatment and control nodes compared to the CasBR approach which implies a higher degree of interference in the experiment.

As our algorithm involves iterating over nodes and their neighbors to identify unassigned nodes, its complexity closely resembles that of a Breadth-First Search (BFS) algorithm. For a graph with $|\mathbf{V}|$ nodes and $|\mathbf{E}|$ edges, the computational complexity of our algorithm is $O(|\mathbf{V}|+|\mathbf{E}|)$. In the worst-case scenario, the graph is a clique, and the complexity of the algorithm is dominated by the edge size which is $O(|\mathbf{V}|^2)$.

Experiments

In this section, we evaluate the performance of different methods in estimating causal effects. We first describe datasets used in our experiments and then discuss our baselines and results.

Semi-Synthetic Data Generation

Since we do not have the underlying ground truth of the causal effects in real-world datasets, we rely on synthetic and semi-synthetic data generation for the evaluation of different network experiment designs.

Dataset Description In our experiments, we consider 10 real-world datasets some of which are attributed. Table 1 represents the datasets' characteristics. For weakly-connected networks, we compute the maximal shortest path as the network's diameter. The *Pubmed* datasets is a citation network with TF/IDF weighted word vectors attributes

Dataset	Nodes	Edges	Attributes	Diameter
Pubmed	19,717	44,325	500	17
Hateful Users	20,810	325,007	523	11
Facebook	4,039	88,234	-	8
Hamsterster	2,059	10,943	6	10
ArXiv	5,242	14,496	-	17
ACM	3,025	13,128	1,870	20
Wiki	2,405	17,981	4,973	9
AMAP	7,650	119,081	745	11

Table 1: Number of nodes, edges, attributes, and diameter of the real-world networks

(Sen et al. 2008). The *Hateful Users* dataset is a subset of Twitter's retweet graph, consisting of users annotated as hateful—i.e., users who post hateful and offensive content on social media—or not, as described in Ribeiro et al. (Ribeiro et al. 2018). *Facebook* ego-network contains social circles on Facebook collected from survey participants using a Facebook app (Leskovec and Mcauley 2012). *Hamsterster* dataset includes the online social network of a group of hamsters (Zheleva et al. 2008). *ArXiv* dataset contains the collaborations between authors who submitted their papers to the General Relativity and Quantum Cosmology category of e-print arXiv (Leskovec, Kleinberg, and Faloutsos 2007). *ACM* dataset is a paper authorship network from the ACM dataset with bag-of-words attributes for each paper. *Wiki* dataset is a word co-occurrence network constructed from the entire set of English Wikipedia pages (Cucerzan 2007). *AMAP* is a product co-purchased network extracted from Amazon with product reviews encoded by bag-of-words as the attributes (Shchur et al. 2018).

We also generate two networks with 5,000 nodes using two different synthetic graph generators. In the *Barabási-Albert* model which generates random scale-free networks with preferential attachment, we set the parameter that controls the number of nodes a new node can attach to 3. In the *Fire-Forest* model, a new node v_i attaches to an existing node v_j and then begins burning links outward from v_j, linking with forward (p_f) and backward (p_b) probabilities to any new node it discovers (Leskovec, Kleinberg, and Faloutsos 2007). In our experiments, we set $p_f = 0.35$ and $p_b = 0.4$.

Cascade Seed Selection In cascade propagation, the seed selection process plays a crucial role, as the configuration of the propagation model, and the size of the cascade are intricately linked to the initial seed nodes' choice. We consider two methods to choose the cascade seed nodes: 1) *random sampling* where a random subset of nodes are selected independently (Poiitis, Vakali, and Kourtellis 2021), and 2) *NewGreedyIC* algorithm, a greedy influence maximization approach to find the smallest set of cascade seed nodes that could maximize the spread of influence in the network (Chen, Wang, and Yang 2009). The NewGreedyIC algorithm is a plausible way that viral marketing campaigns might use to choose their seed nodes with the goal of maximizing their product adoption. In the NewGreedyIC algorithm, we run 100 simulations and set the propagation prob-

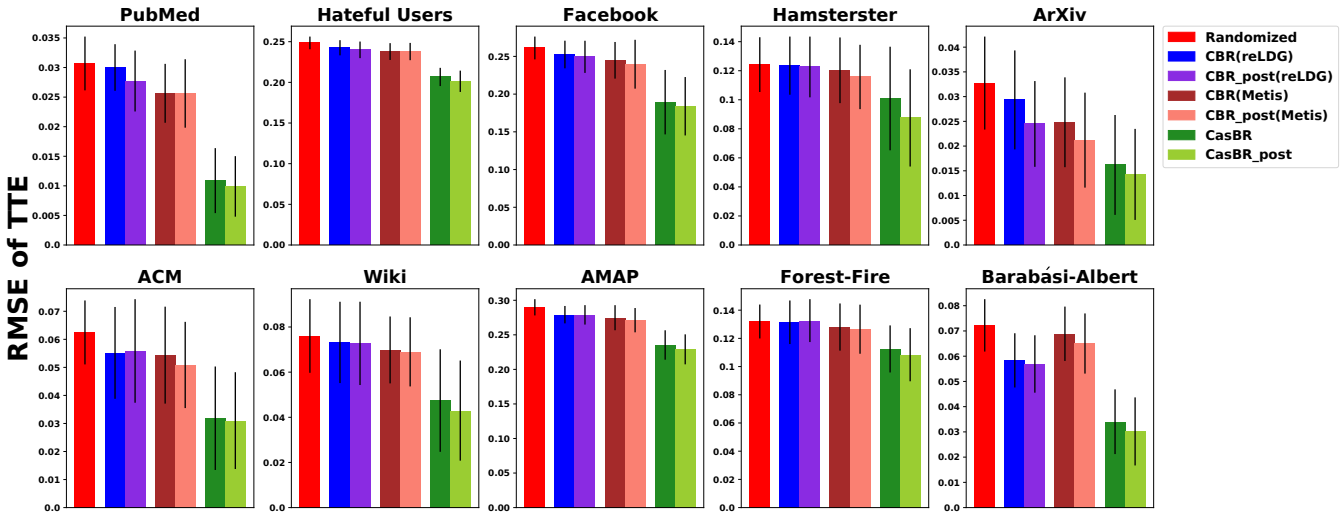


Figure 2: RMSE of total treatment effect estimation of different methods on real-world and synthetic datasets in the last time step of the cascade propagation; Bars show the standard deviation of the estimated effects. Seed nodes are selected using the random sampling method. In all datasets, 10% of the nodes are considered as the cascade seeds. CasBR and CasBR-post achieve the best performance in all datasets.

ability to 0.01.

True Total Treatment Effect Estimation To accurately measure the true underlying TTE, we consider two parallel universes, one where all nodes are receiving treatment and the other where all nodes are in the control group. Initially, we activate the cascade seed nodes and allow their activations to spread through the network. At each step, a newly activated node can activate its neighbors with probability p_{t-t} and p_{c-c} in the treatment and control universes, respectively. This process continues until all m-hop neighbors of each seed node are examined. We then calculate the difference between the average outcomes of nodes in the two universes over Q simulations, which indicates the true underlying TTE.

To quantify the strength of interference, we measure the difference between the average outcomes of treatment and control nodes in each time step of the cascade over Q simulations as estimated TTE in that time step. We report *Root Mean Square Error* of total treatment effects measured as:

$$RMSE = \sqrt{\frac{1}{Q}\sum_{q=1}^{Q}(\hat{\tau}_q - \tau_q)^2} \tag{8}$$

where τ_q and $\hat{\tau}_q$ are the true and estimated TTE in simulation q, respectively. In our cascade generation, we set $Q = 100$, $p_{t-t} = p_{t-c} = 0.05$, and $p_{c-c} = p_{c-t} = 0.02$. The experiments were conducted on an Ubuntu 20.04.4 server equipped with 128 CPUs, each running at 2000 MHz, and 256 GB of RAM.

Baselines

We compare the causal effect estimation error of five different approaches in our experiments:

Dataset	CasBR-post	CBR-post(reLDG)
Pubmed	11.6	23.1
Hateful Users	6.3	8.5
Facebook	17	28.2
Hamsterster	7.1	8.2
ArXiv	4.3	10.4
ACM	2.3	6.2
Wiki	7.5	11.6
AMAP	5.9	7.8
Barabási-Albert	10.5	12.9
Forest-Fire	7.6	10.8

Table 2: Percentage of nodes excluded from the experiments using CasBR-post and CBR-post(reLDG) models

- **Randomized**: In this approach, nodes are randomly assigned to treatment or control independently.
- **CBR**: In the Cluster-based Randomization (CBR) approach, a graph clustering algorithm is deployed to partition the graph nodes into densely-connected clusters, and treatment assignment at the cluster level dictates the node assignment within each cluster (Saveski et al. 2017). We leverage two different graph clustering algorithms: 1) *Restreaming Linear Deterministic Greedy (reLDG)*, which generates a balanced proportion of nodes across partitions (Nishimura and Ugander 2013), and 2) *METIS*, which is a widely-used batch graph clustering algorithm that generates clusters with a minimal number of cross-partition edges (Karypis and Kumar 1998). Both algorithms allow the experimenter to specify the number of clusters. reLDG has been used in prior studies to mitigate interference in network experiments (Saveski

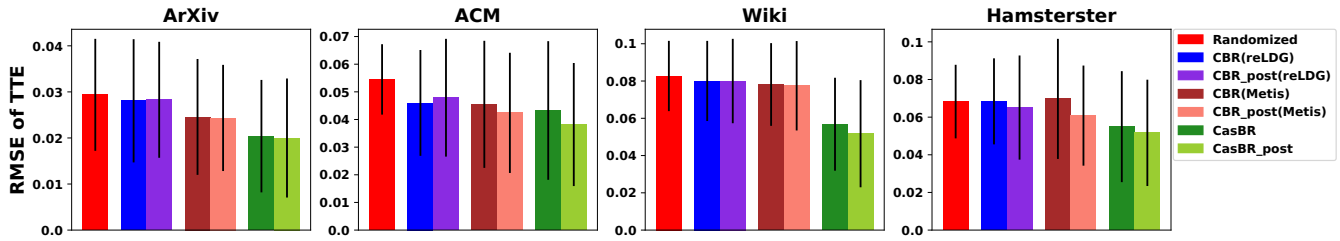
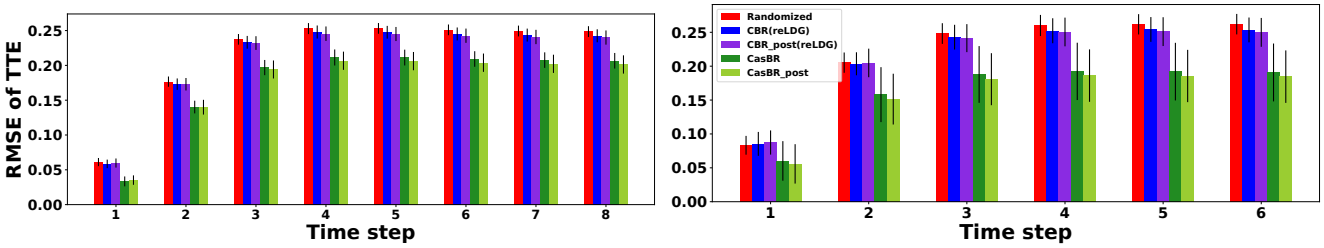


Figure 3: RMSE of total treatment effect estimation of different methods in the last time step of the cascade propagation; Bars show the standard deviation of the estimated effects. Seed nodes are selected using the random sampling method. In all datasets, 10% of the nodes are considered as the cascade seeds. We set $p_{t-t} = p_{t-c} = 0.07$, and $p_{c-c} = p_{c-t} = 0.05$.



(a) Hateful Users

(b) Facebook

Figure 4: RMSE of total treatment effect estimation of different methods in consecutive time steps of the cascade propagation; Bars show the standard deviation of the estimated effects. Seed nodes are selected using the random sampling method. CasBR and CasBR-post get the lowest estimation error in different time steps.

et al. 2017; Pouget-Abadie et al. 2019). In our experiments, CBR(reLDG) and CBR(METIS) represent CBR approaches utilizing reLDG and METIS clustering algorithms, respectively.

- **CBR-post**: This approach is a variant of the CBR method, but some nodes are excluded from the experiment using the post-processing technique described in Algorithm 2. In our experiments, CBR-post(reLDG) and CBR-post(METIS) refer to CBR-post approaches using reLDG and METIS clustering algorithms, respectively.
- **CasBR**: This method is our proposed approach described in Algorithm 1.
- **CasBR-post**: This is a variant of the CasBR approach where the post-processing technique described in Algorithm 2 is applied to the output of the CasBR method.

In CBR and CBR-post methods with both METIS and reLDG algorithms, we set the number of clusters equal to the number of cascade seed nodes in each experiment.

We chose the alphas that led to the lowest RMSE, $\alpha = 0.01$ in all datasets except Facebook where $\alpha = 0.1$ The percentage of nodes excluded from the experiments during post-processing of each dataset using CasBR and CBR(reLDG) is shown in Table 2. We observe that CBR(reLDG) has a higher percentage of nodes with a more similar number of treatment and control neighbors than CasBR, indicating a higher level of interference in the CBR(reLDG) approach. It is worth noting that our causal effect estimator only takes into account the treatment and control nodes. However, we do allow for the peer effects of bystander nodes on treatment and control nodes during TTE estimation. We set the unallowable peer effects from each bystander node to treatment or control nodes to 0.02.

Results

Interference Evaluation. We evaluate the performance of different methods in mitigating interference. Fig. 2 presents the comparison between RMSE of total treatment effects estimated by different frameworks in the last time step of the cascade propagation in all datasets. In this experiment, we use random sampling to select 10% of the nodes as the cascade seed nodes. As expected, the CBR(reLDG) and CBR(METIS) methods outperform the Randomized approach in all datasets, except in the ACM dataset. The results suggest that the estimation error for CBR(reLDG) and CBR(METIS) is similar, except in the case of the Barabási-Albert network, where CBR(reLDG) outperforms CBR(METIS). Furthermore, we observed that CasBR and CasBR-post produced significantly lower estimation errors compared to the baseline methods in all datasets, especially in Pubmed, Hateful Users, Facebook, Arxiv, Wiki, and Barabási-Albert networks, with estimated error reductions of 63.7%, 14.7%, 21.5%, 39.8%, 35.2%, and 33%, respectively, compared to the CBR(reLDG) approach. These findings remained consistent when varying the probabilities of spillover between treatment and control, as depicted in Fig. 3.

We also evaluate the percentage of edges connecting treatment and control nodes as an indicator of interference between these two groups. In this experiment, a random sam-

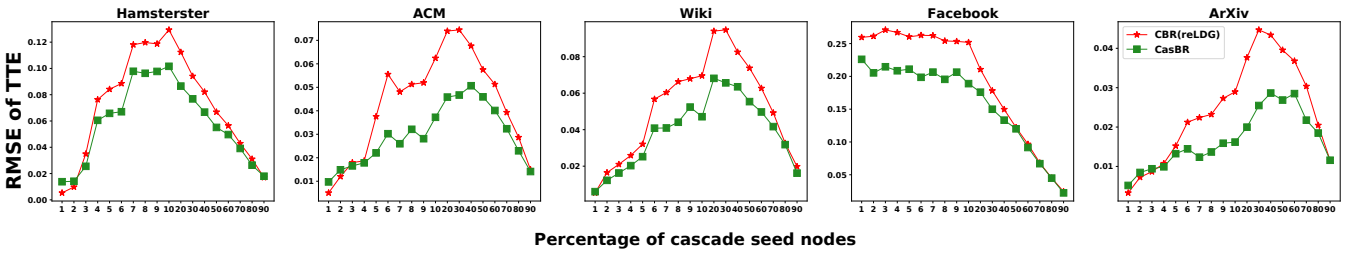


Figure 5: Comparison between RMSE of total treatment effect estimation of CasBR and CBR(reLDG); cascade seed nodes are selected using the random sampling approach. The number of clusters in CBR(reLDG) is equal to the number of cascade seeds.

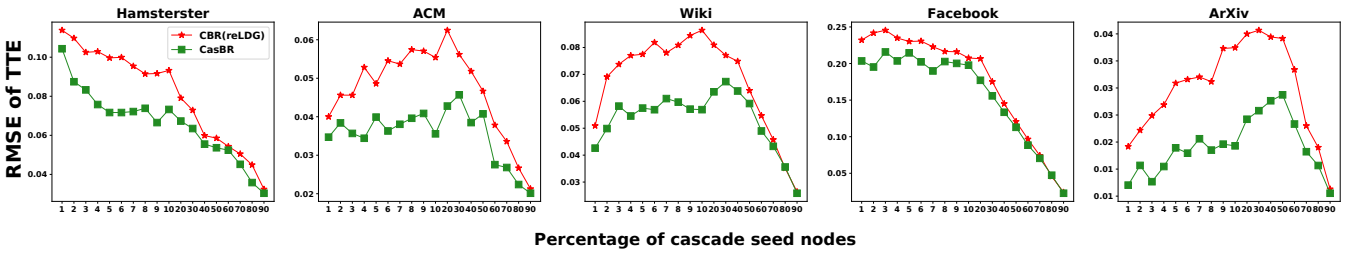


Figure 6: Comparison between RMSE of total treatment effect estimation of CasBR and CBR(reLDG); cascade seed nodes are selected using the NewGreedyIC approach. The number of clusters in CBR(reLDG) is equal to the number of cascade seeds.

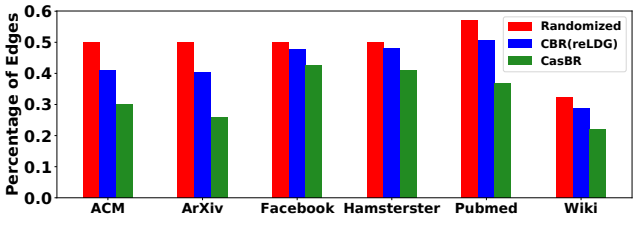


Figure 7: Percentage of edges between treatment and control nodes. In all datasets, 10% of the nodes are considered as the cascade seeds.

pling technique is employed to choose 10% of graph nodes as the seeds for the cascade. As illustrated in Fig. 7, CasBR consistently demonstrates the lowest number of edges between treatment and control nodes across all datasets. In contrast, both the Randomized and CBR(reLDG) methods consistently exhibit the highest number of such edges.

Results show that applying the post-processing algorithm on CasBR, CBR(reLDG), and CBR(METIS) can improve the causal effect estimation error in all datasets. For example, the post-processing method reduces the estimation error of CasBR by 8% in Hamsterster, 11.8% in Arxiv, 9.3% in Wiki, and 8.3% in Barabási-Albert networks, and the estimation error of CBR(reLDG) by 7.6% in PubMed and 2.2% in Facebook dataset. Given the similar performance of CBR(reLDG) and CBR(METIS), we use CBR(reLDG) exclusively in the subsequent experiments.

Interference Evaluation Over Time. In this experiment, we assess the estimation error of CasBR and CBR(reLDG) in different time steps of cascade propagation. As represented in Fig 4, the interference increases significantly over the cascade growth, with Randomized yielding an increase from 0.06 to 0.24, CBR(reLDG) from 0.05 to 0.24, and CasBR from 0.03 to 0.2 in the Hateful Users dataset, and Randomized yielding an increase from 0.08 to 0.26, CBR(reLDG) from 0.08 to 0.24, and CasBR from 0.06 to 0.19 in the Facebook dataset. Nonetheless, our experiments on both datasets show that CasBR and CasBR-post consistently outperform other baselines in all time steps of the cascade propagation. For instance, in the first time step of the cascade in the Hateful Users dataset, CasBR reduces the estimation error by 42.9% compared to CBR(reLDG), and this reduction increases to 14.8% in the last time step. The findings hold true for the Facebook dataset.

Sensitivity to the Number of Cascade Seed Nodes. In this experiment, we explore the estimation error of CasBR vs. CBR(reLDG) when varying the number of cascade seed nodes. Considering the random sampling approach, Fig. 5 represents that CasBR gets a better performance compared to CBR(reLDG) in all datasets. In addition, by increasing the number of cascade seeds which represents the number of clusters in the CBR(reLDG) method, the difference between the estimation error of CasBR and CBR(reLDG) increases, especially in ACM and ArXiv datasets. Moreover, by increasing the number of cascade seeds, the estimation error for TTE in both methods follows a pattern of initial increase followed by a subsequent decrease, except in the Facebook dataset where it shows a decrease in the estimation error. For a high number of cascade seed nodes, the network quickly saturates, leading to the eventual convergence of error rates for both methods.

In contrast to employing random sampling, we employ

the NewGreedyIC algorithm for the purpose of selecting the initial nodes for the cascade. Our results, presented in Fig. 6, align with the previous findings that CasBR outperforms CBR(reLDG). However, in the Hamsterster dataset, we observe a decreasing trend in TTE estimation error when increasing the number of cascade seed nodes. These results highlight the impact of network properties on the amount of interference between treatment and control nodes, particularly in the context of selecting influential nodes. The set of influential nodes selected by the NewGreedyIC algorithm varies for different percentages, and increasing the percentage may exclude certain nodes from the set. Therefore, a model's performance may be influenced by the network structure and ego network of the influential nodes.

Discussion

Cascades of influence are natural phenomena frequently encountered in social systems experimentation. While previous studies have often advocated for clustering as the preferred method to mitigate interference in network experiments, this paper questions this approach by demonstrating that clustering may not be the best strategy when dealing with cascades of influence. Instead, we propose a network experiment design, CasBR, that leverages knowledge about cascade seed nodes to propagate treatment assignments to their multi-hop neighbors, thus ensuring that both the seeds and a significant portion of their multi-hop neighbors belong to the same treatment group.

Our empirical results demonstrate the better performance of CasBR compared to the baselines. While in some datasets, e.g., PubMed and ArXiv, the RMSE seems to be small, it is important to also consider the effect size, because even a small error in the estimation can result in suboptimal decision-making. In large-scale marketing experiments, small effects (e.g., $< 1\%$) can have significant implications for decision-making (Blake and Coey 2014; Fradkin and Holtz 2023). A low RMSE (e.g., 0.01) indicates precise predictions, facilitating the detection and estimation of small effects. However, if the RMSE increases (e.g., 0.03), model accuracy decreases, and small effects may be suppressed by noise, leading to potential misinterpretation and missed opportunities for improvement.

Additionally, our findings highlight the impact of the number of active cascade seed nodes on the accuracy of causal effect estimation across various methods. We observe that by increasing the number of cascade seeds, the causal effect estimation error in both methods follows a pattern of initial increase followed by a subsequent decrease. However, as the number of activated cascade seed nodes increases substantially, the network experiences rapid saturation. This saturation ultimately leads to the convergence of performance between CBR and CasBR methods.

It is important to acknowledge that our proposed method relies on the availability of information about cascade seed nodes. This is true, for example, when we know the early adopters of a product (e.g., people who bought the newest iPhones). However, when such information is not unknown and cannot be easily inferred, our approach would not be applicable.

One natural extension of this work is the development of network experiment frameworks for other models of information diffusion, such as the linear threshold model. Another possible extension is designing an experiment for scenarios in which the likelihood of neighbor activations can be inferred from cascades that already have been observed in the network. Addressing selection bias, i.e., the problem that treatment and control nodes can represent different populations of individuals (Fatemi and Zheleva 2020a), in network experiments with cascades is also a fruitful future direction.

Acknowledgments

This material is based on research sponsored in part by NSF under grant No. 2047899, and DARPA under contract number HR001121C0168.

References

Antman, E. M.; Lau, J.; Kupelnick, B.; Mosteller, F.; and Chalmers, T. C. 1992. A comparison of results of meta-analyses of randomized control trials and recommendations of clinical experts: treatments for myocardial infarction. *Jama*, 268(2): 240–248.

Aral, S.; and Walker, D. 2011. Creating social contagion through viral product design: A randomized trial of peer influence in networks. *Management science*, 57(9): 1623–1639.

Aronow, P. M.; and Middleton, J. A. 2013. A class of unbiased estimators of the average treatment effect in randomized experiments. *Journal of Causal Inference*, 1(1): 135–154.

Aronow, P. M.; and Samii, C. 2017. Estimating average causal effects under general interference, with application to a social network experiment. *The Annals of Applied Statistics*, 11(4): 1912–1947.

Backstrom, L.; and Kleinberg, J. 2011. Network bucket testing. In *WWW*.

Bailey, M.; Johnston, D. M.; Kuchler, T.; Stroebel, J.; and Wong, A. 2019. Peer effects in product adoption. Technical report, National Bureau of Economic Research.

Bakshy, E.; Eckles, D.; Yan, R.; and Rosenn, I. 2012. Social influence in social advertising: evidence from field experiments. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, 146–161. ACM.

Basse, G. W.; and Airoldi, E. M. 2015. Optimal design of experiments in the presence of network-correlated outcomes. *ArXiv e-prints*, 6994539(10.1093).

Betts, J. A.; Richardson, J. D.; Chowdhury, E. A.; Holman, G. D.; Tsintzas, K.; and Thompson, D. 2014. The causal role of breakfast in energy balance and health: a randomized controlled trial in lean adults. *The American journal of clinical nutrition*, 100(2): 539–547.

Bharathi, S.; Kempe, D.; and Salek, M. 2007. Competitive influence maximization in social networks. In *Internet and Network Economics: Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007. Proceedings 3*, 306–311. Springer.

Blake, T.; and Coey, D. 2014. Why marketplace experimentation is harder than it seems: The role of test-control interference. In *Proceedings of the fifteenth ACM conference on Economics and computation*, 567–582.

Brennan, J. R.; Mirrokni, V.; and Pouget-Abadie, J. 2022. Cluster Randomized Designs for One-Sided Bipartite Experiments. In *Advances in Neural Information Processing Systems*.

Candogan, O.; Chen, C.; and Niazadeh, R. 2021. Correlated cluster-based randomized experiments: Robust variance minimization. *Chicago Booth Research Paper (21-17)*.

Chen, D.-B.; Gao, H.; Lü, L.; and Zhou, T. 2013. Identifying influential nodes in large-scale directed networks: the role of clustering. *PloS one*, 8(10): e77455.

Chen, W.; Wang, Y.; and Yang, S. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 199–208.

Cheng, J.; Adamic, L.; Dow, P. A.; Kleinberg, J. M.; and Leskovec, J. 2014. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, 925–936.

Choi, D. 2017. Estimation of monotone treatment effects in network experiments. *Journal of the American Statistical Association*, 112(519): 1147–1155.

Cucerzan, S. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 708–716.

Daneshmand, H.; Gomez-Rodriguez, M.; Song, L.; and Schoelkopf, B. 2014. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *International conference on machine learning*, 793–801. PMLR.

Eckles, D.; Karrer, B.; and Ugander, J. 2016. Design and Analysis of Experiments in Networks: Reducing Bias from Interference. *Journal of Causal Inference*, 5(1): 20150021.

Eckles, D.; Karrer, B.; and Ugander, J. 2017. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1).

Eckles, D.; Kizilcec, R.; and Bakshy, E. 2016. Estimating peer effects in networks with peer encouragement designs. *PNAS*.

Fatemi, Z.; and Zheleva, E. 2020a. Minimizing Interference and Selection Bias in Network Experiment Design. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1): 176–186.

Fatemi, Z.; and Zheleva, E. 2020b. Network Experiment Design for Estimating Direct Treatment Effects. In *KDD Workshop on Mining and Learning with Graphs (MLG)*, volume 8.

Fatemi, Z.; and Zheleva, E. 2023. Network experiment designs for inferring causal effects under interference. *Frontiers in big Data*, 6: 1128649.

FORCE11. 2020. The FAIR Data principles. https://force11.org/info/the-fair-data-principles/. Accessed: 2023-07-01.

Fradkin, A.; Grewal, E.; and Holtz, D. 2021. Reciprocity and unveiling in two-sided reputation systems: Evidence from an experiment on Airbnb. *Marketing Science*, 40(6): 1013–1029.

Fradkin, A.; and Holtz, D. 2023. Do incentives to review help the market? evidence from a field experiment on airbnb. *Marketing Science*.

Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.

Glanz, J. M.; Wagner, N. M.; Narwaney, K. J.; Kraus, C. R.; Shoup, J. A.; Xu, S.; O'Leary, S. T.; Omer, S. B.; Gleason, K. S.; and Daley, M. F. 2017. Web-based social media intervention to increase vaccine acceptance: a randomized controlled trial. *Pediatrics*, 140(6).

Gomez-Rodriguez, M.; Leskovec, J.; and Krause, A. 2012. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4): 1–37.

Halloran, M. E.; and Hudgens, M. G. 2012. Causal inference for vaccine effects on infectiousness. *The international journal of biostatistics*, 8(2): 1–40.

Halloran, M. E.; and Struchiner, C. J. 1995. Causal inference in infectious diseases. *Epidemiology*, 142–151.

Harshaw, C.; Sävje, F.; Eisenstat, D.; Mirrokni, V.; and Pouget-Abadie, J. 2021. Design and analysis of bipartite experiments under a linear exposure-response model. *arXiv preprint arXiv:2103.06392*.

Ho, D. E. 2017. Does peer review work? An experiment of experimentalism. *Stan. L. Rev.*, 69: 1.

Holtz, D.; Lobel, R.; Liskovich, I.; and Aral, S. 2020. Reducing interference bias in online marketplace pricing experiments. *Available at SSRN 3583836*.

Hudgens, M. G.; and Halloran, M. E. 2008. Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482): 832–842.

Imbens, G.; and Rubin, D. 2015. *Causal Inference in Statistics, Social and Biomedical Sciences: An Introduction*. Cambridge Univ Press.

Johari, R.; Li, H.; Liskovich, I.; and Weintraub, G. Y. 2022. Experimental design in two-sided platforms: An analysis of bias. *Management Science*, 68(10): 7069–7089.

Karypis, G.; and Kumar, V. 1998. Multilevelk-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed computing*, 48(1): 96–129.

Kazdin, A. E. 2022. Drawing causal inferences from randomized controlled trials in psychotherapy research. *Psychotherapy Research*, 1–13.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 137–146.

Kohavi, R.; and Longbotham, R. 2017. Online Controlled Experiments and A/B Testing. *Encyclopedia of machine learning and data mining*, 7(8): 922–929.

Leskovec, J.; Kleinberg, J.; and Faloutsos, C. 2007. Graph Evolution: Densification and Shrinking Diameters. *ACM Trans. Knowl. Discov. Data*, 1(1).

Leskovec, J.; and Mcauley, J. 2012. Learning to discover social circles in ego networks. *Advances in neural information processing systems*, 25.

Li, C.; Ma, J.; Guo, X.; and Mei, Q. 2017. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th international conference on World Wide Web*, 577–586.

Morone, F.; and Makse, H. A. 2015. Influence maximization in complex networks through optimal percolation. *Nature*, 524(7563): 65–68.

Netrapalli, P.; and Sanghavi, S. 2012. Learning the graph of epidemic cascades. *ACM SIGMETRICS Performance Evaluation Review*, 40(1): 211–222.

Nishimura, J.; and Ugander, J. 2013. Restreaming graph partitioning: simple versatile algorithms for advanced balancing. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1106–1114.

Poiitis, M.; Vakali, A.; and Kourtellis, N. 2021. On the Aggression Diffusion Modeling and Minimization in Twitter. *ACM Trans. Web*, 16(1).

Pouget-Abadie, J.; and Horel, T. 2015. Inferring graphs from cascades: A sparse recovery framework. In *International Conference on Machine Learning*, 977–986. PMLR.

Pouget-Abadie, J.; Mirrokni, V.; Parkes, D. C.; and Airoldi, E. M. 2018. Optimizing Cluster-Based Randomized Experiments under Monotonicity. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, 2090–2099. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355520.

Pouget-Abadie, J.; Saint-Jacques, G.; Saveski, M.; Duan, W.; Ghosh, S.; Xu, Y.; and Airoldi, E. M. 2019. Testing for arbitrary interference on experimentation platforms. *Biometrika*, 106(4): 929–940.

Ribeiro, M. H.; Calais, P. H.; Santos, Y. A.; Almeida, V. A.; and Meira Jr, W. 2018. Characterizing and Detecting Hateful Users on Twitter. *ICWSM*.

Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5): 688.

Saint-Jacques, G.; Varshney, M.; Simpson, J.; and Xu, Y. 2019. Using ego-clusters to measure network effects at LinkedIn. *arXiv preprint arXiv:1903.08755*.

Saveski, M.; Pouget-Abadie, J.; Saint-Jacques, G.; Duan, W.; Ghosh, S.; Xu, Y.; and Airoldi, E. 2017. Detecting network effects: Randomizing over randomized experiments. In *KDD*.

Sen, P.; Namata, G. M.; Bilgic, M.; Getoor, L.; Gallagher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI Magazine*, 29(3): 93–106.

Shakya, H. B.; Stafford, D.; Hughes, D. A.; Keegan, T.; Negron, R.; Broome, J.; McKnight, M.; Nicoll, L.; Nelson, J.; Iriarte, E.; et al. 2017. Exploiting social influence to magnify population-level behaviour change in maternal and child health: study protocol for a randomised controlled trial of network targeting algorithms in rural Honduras. *BMJ open*, 7(3): e012996.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *NeurIPS-W*.

Stuart, E. A. 2010. Matching methods for causal inference: A review and a look forward. *Statistical science*, 25(1): 1.

Tchetgen, E. J. T.; and VanderWeele, T. J. 2012. On causal inference in the presence of interference. *Statistical methods in medical research*, 21(1): 55–75.

Toulis, P.; and Kao, E. 2013. Estimation of causal peer influence effects. In *ICML*, 1489–1497.

Ugander, J.; Karrer, B.; Backstrom, L.; and Kleinberg, J. 2013. Graph cluster randomization: Network exposure to multiple universes. In *KDD*.

Ugander, J.; and Yin, H. 2020. Randomized graph cluster randomization. *arXiv preprint arXiv:2009.02297*.

Ugander, J.; and Yin, H. 2023. Randomized graph cluster randomization. *Journal of Causal Inference*, 11(1): 20220014.

Wang, C.; Chen, W.; and Wang, Y. 2012. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery*, 25(3): 545–576.

Zhang, J.; Brackbill, D.; Yang, S.; Becker, J.; Herbert, N.; and Centola, D. 2016. Support or competition? How online social networks increase physical activity: A randomized controlled trial. *Preventive Medicine Reports*, 4: 453–458.

Zhao, Y.; Li, S.; and Jin, F. 2016. Identification of influential nodes in social networks with community structure based on label propagation. *Neurocomputing*, 210: 34–44. SI:Behavior Analysis In SN.

Zheleva, E.; J.; Kuter, U.; and Getoor, L. 2008. Using Friendship Ties and Family Circles for Link Prediction. *SNAKDD*.

Paper Checklist

1. For most authors...

   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Yes

   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes

(c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes

(d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? Yes

(e) Did you describe the limitations of your work? Yes

(f) Did you discuss any potential negative societal impacts of your work? Yes

(g) Did you discuss any potential misuse of your work? Yes

(h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? Yes

(i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes

2. Additionally, if your study involves hypotheses testing...

(a) Did you clearly state the assumptions underlying all theoretical results? NA

(b) Have you provided justifications for all theoretical results? NA

(c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? NA

(d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? NA

(e) Did you address potential biases or limitations in your theoretical framework? NA

(f) Have you related your theoretical results to the existing literature in social science? NA

(g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? NA

3. Additionally, if you are including theoretical proofs...

(a) Did you state the full set of assumptions of all theoretical results? NA

(b) Did you include complete proofs of all theoretical results? NA

4. Additionally, if you ran machine learning experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes

(e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes

(f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? NA

5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity**...

(a) If your work uses existing assets, did you cite the creators? Yes

(b) Did you mention the license of the assets? NA

(c) Did you include any new assets in the supplemental material or as a URL? NA

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? NA

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? Yes

(f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see FORCE11 (2020))? NA

(g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see Gebru et al. (2021))? NA

6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity**...

(a) Did you include the full text of instructions given to participants and screenshots? NA

(b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? NA

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA

(d) Did you discuss how data is stored, shared, and deidentified? NA

Ethics Statement

This research project was conducted in full compliance with research ethics norms and guidelines. We take ethical considerations seriously and have taken the following steps to ensure that our research is conducted in an ethical and responsible manner:

- Participant Privacy: We have respected the privacy of individuals and organizations included in our study. We have used publicly available datasets and synthetic data to minimize the possibility of revealing any sensitive information. We have also followed the terms and conditions of online social media platforms and have not published any information or links that may lead to individual identification.
- Informed Consent: As our research does not involve direct interaction with people, we did not require informed consent from human participants. However, we have ensured that the use of publicly available datasets is in accordance with the terms of use set by the data sources.

- Data Sharing: We utilized publicly available datasets, with corresponding dataset references incorporated into the text. Upon acceptance of the paper, we will make the code available along with comprehensive instructions for its execution.
- Conflicts of Interest: We declare no conflicts of interest that could have influenced the results of our study. We have conducted our research objectively and without any external influence.

We believe that our research method does not pose any ethical concerns, and we have taken appropriate steps to ensure that our research is conducted ethically and responsibly. Unless our experiment design is used to test something unethical, we do not foresee negative societal impacts. It is important to note that in our experiments, we employed unsupervised Machine Learning techniques (clustering), not classification.