Causal Network Motifs: Identifying Heterogeneous Spillover Effects in A/B Tests

Yuan Yuan
Massachusetts Institute of Technology
yuan2@mit.edu

Kristen M. Altenburger
Facebook Core Data Science
kaltenburger@fb.com

Farshad Kooti
Facebook Core Data Science
farshadkt@fb.com

ABSTRACT

Randomized experiments, or "A/B" tests, remain the gold standard for evaluating the causal effect of a policy intervention or product change. However, experimental settings such as social networks, where users are interacting and influencing one another, violate conventional assumptions of no interference needed for credible causal inference. Existing solutions include accounting for the fraction or count of treated neighbors in a user's network, among other strategies. Yet, most current methods do not account for the local network structure beyond simply counting the number of neighbors. Capturing local network structures is important because it can account for theories, such as structural diversity and echo chambers. Our study provides an approach that accounts for both the local structure in a user's social network via motifs as well as the assignment conditions of neighbors. We propose a two-part approach. We first introduce and employ "causal network motifs", which are network motifs that characterize the assignment conditions in local ego networks; and then we propose a tree-based algorithm for identifying different network interference conditions and estimating their average potential outcomes. We test our method on a real-world experiment on a large-scale network and a synthetic network setting, which highlight how accounting for local structures can better account for different interference patterns in networks.

ACM Reference Format:
Yuan Yuan, Kristen M. Altenburger, and Farshad Kooti. 2021. Causal Network Motifs: Identifying Heterogeneous Spillover Effects in A/B Tests. In *Proceedings of ACM Conference (WWW'21).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Randomized control trials, or "A/B tests", have been fundamental to understanding the impact of an intervention, such as a new policy [34, 36], product intervention [7, 13], or medical treatment [5, 44]. A/B tests can estimate the causal effect of a treatment intervention by ensuring that treatment and control assignments are independent of other variables. Increasingly, causal inference methods have had to adapt to modern A/B test settings where there are high-dimensional features [26, 28, 49, 56], computational and algorithmic considerations [69], and network interference concerns [6, 27, 55].

Traditional causal inference methods rely on a critical assumption called the "stable unit treatment value assumption" (SUTVA) [29, 59]. SUTVA is in fact a strong assumption, requiring that a unit's outcome is only affected by its own assignment conditions, regardless of the assignment conditions of all other observations. However, this can be an unrealistic assumption in many settings such as social networks where user's are influenced by one another [11, 17, 46], which we refer to here as *network interference* [6].

An increasing number of approaches are aimed at dealing with network interference [8, 15, 37, 40, 50, 55, 61, 63, 70]. Existing methods for addressing networked interference can be categorized into two main strategies. The first one is to improve the random assignment strategy. Cluster random assignment treats observations at the level in which observations have strong interdependence (e.g., assigning on the class or city level) [15, 33, 50]. Graph cluster randomization is a special case for social networks [27, 63]. It first runs a graph clustering algorithm and then randomizes treatment assignment on the graph cluster level. Recent work proposes random clustering instead of a fixed clustering [64]. Another type of approach, including ours, relaxes SUTVA by allowing the potential outcome to be defined as a function of the assignment conditions of the ego observation and its (n-hop) neighbors [42, 65]. However, few studies have utilized the network structure in the n-hop neighbors (i.e. local network structure) to further characterize different interference conditions.

From the empirical end, many social network studies have highlighted why local structures should be considered to address network interference. For example, the structural diversity hypothesis [60, 62] claims the likelihood of product adoption is largely dependent on the degree to which a unit's neighbors who have adopted are disconnected. By contrast, complex contagion theory [18, 19] or the "echo chamber" effect [14, 30], suggest an individual is most likely to adopt a behavior when she is clustered in the network of multiple neighbors who have adopted this behavior. Figure 1 illustrates four examples of network interference that could be captured by the local networks structure and treatment assignment. The first two conditions are simply the cases where all neighbors are treated or non-treated, followed by the important network interference conditions suggested by structural diversity and complex contagion, respectively. In the case of structural diversity and echo chamber settings, the ego node in c) and d) has 1/2 neighbors treated but exhibit very different local structures and the ego's outcome may be different in these settings; and we do not know which one is the dominant factor that drives most variance in the outcome.

Our study provides a tool for experimenters and practitioners to account for important network interference conditions without necessarily specifying a particular dominant social science theory. We rely on network motifs [3, 45] (or graphlets [53]) to characterize

WWW'21, April 2021, Ljubljana, Slovenia
© 2021 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

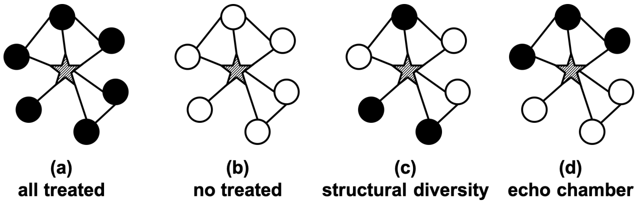

Figure 1: Examples of network interference conditions across different local network structures. The star indicates a user and a circle represents a user's friends. Solid circles indicate that a friend is in treatment and hollow circles indicate a friend is in control. For stars, the shaded indicates that it could be treated or control.

both the local structure and treatment assignment conditions to account for the different types of (network interference) exposure conditions. We can think of network motifs as "label-independent structure" since the features do not depend on treatment labels and network motifs with treatment assignment conditions as "label-dependent structure" since they depend on labels [32]. We refer to network motifs with treatment assignment labels as *causal network motifs*. Prior work has considered label-dependent network motifs for use in protein function prediction applications [21] and has developed computationally tractable ways for detecting labeled motifs [48]. Another example is [9], which uses network features such as transitivity in a propensity score matching framework, with a focus on observational data. Our proposal is distinct in that we focus on experimental settings where causal network motifs can quantify the causal impact of different interference conditions.

In this paper, we provide an approach to automatically categorize network interference conditions based on local network neighborhood structures. We develop our approach based on the framework in [8], where the authors regard different network interference conditions[1] as unique treatments. However, it is unclear how experimenters or practitioners should a priori define different network interference conditions that are suitable for specific data and experiment. Moreover, manually determining different network interference conditions may make the issue of a large number of researcher degrees of freedom more severe.

The following sections will describe our two-step solution to identify different network interference conditions. First, we construct network motifs with assignment labels, referred to as *causal network motifs*, to characterize complex network interference conditions. In our study, we not only characterize the network structure by the number of different network motifs, but also characterize the assignment conditions in the local network neighborhood by the causal network motifs. Second, using these network motif features, we develop a tree-based algorithm to cluster different exposure conditions. Each leaf or terminal node in the tree corresponds to a network interference condition and can be used to estimate the average potential outcome given that condition. We cannot directly apply conventional decision trees or other machine learning algorithms, since we need to adjust those algorithms to address issues such as selection bias and positivity as will be explained [22, 23, 68]. Moreover, as a common goal in causal inference, we aim to estimate the average potential outcome given an interference condition to quantify causal impacts, rather than predicting the potential outcome and the treatment effect for every observation [52].

The rest of the paper is organized as follows. Section 2 provides a broad review on the graph mining and causal inference literature. Section 3 provides preliminaries of causal inference, the potential outcomes framework, and why the SUTVA may fail in network experiments. Then we illustrate our two-part approach: Section 4 presents how we construct the "causal network motifs," which characterize the treatment assignment conditions of each observation as well as their network neighborhood; and Section 5 discusses our tree-based algorithm that regards the "causal network motifs" as input and then labels each observation as an exposure condition. Section 6 empirically shows the validity of our approach using a synthetic experiment and a real-world experiment. Section 7 concludes.

2 GRAPH MINING AND CAUSAL INFERENCE

Our work falls at the intersection of graph mining and causal inference. Graph mining methods are aimed at analyzing network properties such as the average number of connections, the number of communities, and more [20, 24]. We will focus on one type of graph property called a motif. A motif is a simple way to characterize subgraph patterns on a set of nodes [2, 45]. Motifs can also account for the node label or attribute [1], which is a type of label-dependent structure that accounts for both the node label and network topology [32]. However, we are not aware of any applications that have used labeled network motifs in causal inference applications, which is the focus of this work. There is also a growing literature on adapting machine learning algorithms for causal inference applications [10, 66]. Our work also builds closely on the causal inference literature and the potential outcomes framework [39, 52] as will be discussed in more detail in the following sections.

3 CAUSAL INFERENCE SETUP

Let i (or j) index individuals in a social network. $\mathcal{U}$ is the set of nodes in the population. $\mathcal{N}_i$ be the neighbor set of i, and this can be extended beyond immediate neighbors. Y_i is the observed potential outcome. $Z_i \in \{0, 1\}$ denotes the random treatment assignment for i. Whenever applicable, we use upper-case letters to represent variables or vectors that could be intervened by experimenters or affected by the intervention; and we use lower-case letters for other variables that are not affected by the intervention. Sets are in calligraphy.

Potential outcomes framework for network interference. We employ the potential outcomes framework for this set-up [52]. The potential outcomes framework defines that a unit's outcome is a function of assignment conditions.

$$Y_i = y_i(Z_i). \tag{1}$$

$y_i(1)$ and $y_i(0)$ are called potential outcomes and we only ever observe one of them. This function implies no interference, meaning the potential outcome Y_i does not depend on the treatment assignment of other users. However, this is often an unrealistic

[1] In this paper, we use exposure conditions and network interference conditions interchangeably.

assumption in network settings, since the unit's outcome can be dependent on the treatment assignment conditions of neighbors or even any other unit, through, for example, social contagion.

To overcome this unrealistic assumption, our approach is based on [8] which introduces exposure mapping. An exposure mapping is defined as a function that maps the treatment assignment vector to a value; the fraction of treated friends is an example of a simple exposure mapping. Each exposure mapping corresponds to a condition that considers both treatment assignment conditions of the ego node and her (n-hop) neighbor. Under this framework, researchers or practitioners usually need to define a priori a set of network interference conditions (also known as *exposure conditions*). We denote this set by $\mathcal{D}$. Then in this framework, the outcome variable is expressed by

$$Y_i = \sum_{d \in \mathcal{D}} y_i(d) \mathbb{1}[D_i = d]. \tag{2}$$

Each d corresponds to a pre-specified exposure condition (interference condition). D_i is a random variable indicating i's exposure condition. Analyzing different exposure conditions can offer insights into how observations react to different interference settings, such as comparing (with selection bias corrected) users in control with treated friends to users in control with only control friends.

Let us introduce three examples. First, SUTVA is a special case under this exposure mapping framework (where $D = \{ego\_treated, ego\_control\}$). Another example is the existence of direct and indirect effects. A simple example introduced in [8] is that $D = \{no\_exposure,\ direct\_exposure, indirect\_exposure, direct + indirect\ \_exposure\}$. Direct effect means the ego or user is treated and indirect effect means any neighbor is treated; thus we have four conditions in total. In our study, we use direct effect to refer to the effect of changing Z_i from 0 to 1 on the outcome variable; and indirect effect refers to any consequence that result from the assignment conditions besides ego node i, i.e. Z_{-i}.[2]

Finally, note that the k (or q-fractional) neighborhood conditions in [63] fits this framework, with (2 × 2) four different exposure conditions — whether the ego node is treated × whether more than k (or q-fractional) of her neighbors are assigned to the same treatment condition with the ego node.

Distinguishing between correlation and causation. The indirect effect may contain a large degree of heterogeneity based on how many (n-hop) neighbors are treated and how they are connected. Some of these conditions are theorized in [19, 62] while other cannot. It is usually challenging to distinguish between what is the causal impact of certain exposure conditions, versus what is usually confounded by ego's local network structure (for example, users with more friends tend to have their friends less clustered) [60]. In other words, selection bias may result from simply taking average over observations within a certain interference condition. For example, imagine that we want to quantify the impact of an indirect effect, so simply taking the average over observations who have at least one treated friend versus taking average over observations with no treated friends. This may be biased towards observations who have more friends. We illustrate this selection bias issue in Figure 4.

Mathematically, our goal is to estimate the average potential outcome ($\bar{y}(d) = \frac{1}{|\mathcal{U}|} \sum_i y_i(d)$, for all $d \in \mathcal{D}$). This is one of the core goals in [8]. To correct the aforementioned selection bias, we employ inverse probability weighting, such as Horvitz–Thompson estimator and Hajek estimator [57] Although the Horvitz–Thompson estimator is an unbiased estimator for the average potential outcome, it empirically has unaffordably high variance. Therefore, following [8, 27], we use the Hajek estimator to estimate the average potential outcome for exposure condition d since its small bias can usually be ignored in a large sample:

$$\hat{\bar{y}}_{Hajek}(d) = \frac{\sum_{i \in \mathcal{U}} \frac{1_i(d)\, y_i(d)}{\pi_i(d)}}{\sum_{i \in \mathcal{U}} \frac{1_i(d)}{\pi_i(d)}} = \frac{\sum_{i \in \mathcal{U}, D_i = d} \frac{y_i(d)}{\pi_i(d)}}{\sum_{i \in \mathcal{U}, D_i = d} \frac{1}{\pi_i(d)}}, \tag{3}$$

where we denote $\pi_i(d) = \mathbb{P}[D_i = d]$ and $1_i(d) = \mathbb{1}[D_i = d]$. $\pi_i(d)$ is the inclusion probability that the exposure condition of i is d, and is a generalization of propensity scores [51]. It can also be understood as a weighted average over observations with $1_i(d) = 1$, where the weight is $1/\pi_i(d)$. Therefore, it can be estimated through weighted linear regressions: the coefficient for the constant is the Hajek estimator. Its variance can be estimated via Taylor linearization [54].

The probability $\pi_i(d)$ is often challenging to compute analytically. We therefore use Monte Carlo for sufficiently large replicates to approximate $\pi_i(d)$. Specifically, we re-run the treatment assignment procedure for R replicates to obtain the empirical distribution of $(Z_i, \mathbf{X_i})$.[3] Therefore, we can derive the estimated inclusion probability $\hat{\pi}_i(d)$ for any exposure condition d. For example, we can let $\hat{\pi}_i(d) = \frac{\sum_r \mathbb{1}[D_i^{(r)} = d] + 1}{R+1}$; and we substitute the $\pi_i(d)$ in Equation 3 with $\hat{\pi}_i(d)$. The relative bias $\hat{\bar{y}}_{Hajek}(d)$ to $\bar{y}(d)$ diminishes exponentially as R increases. The details have been discussed in [8].

While the exposure mapping framework introduces a tractable way to take into account network interference, it is an open question how we should then account for the treatment assignment conditions of the n-hop neighbors and their network connections in Figure 1. We hope to provide an approach that generates exposure conditions that are suitable for a given experiment and dataset, and that avoids manually defining exposure conditions a priori. Our study provides a two-step solution to automatically identify different exposure conditions while overcoming selection bias concerns, as will be explained in more detail in the next sections. First, for an A/B test on a network, we construct network motif features with treatment assignment conditions to provide a fine-grained characterization of the local network structure and potential interference conditions. Second, using the network motif characterization as input, we develop a tree-based algorithm to perform clustering and define the set $\mathcal{D}$ rather than allowing practitioners to explore that.

4 CAUSAL NETWORK MOTIFS

Network motifs are a way to characterize all patterns of smaller network features among a set of nodes [45]. We introduce "causal network motifs" which differ from conventional network motifs

[2] Indirect effects will be further specified by $\mathbf{X}_i$ in later sections.

[3] It is challenging to derive this distribution analytically, especially when the random assignment is deployed on the individual level.

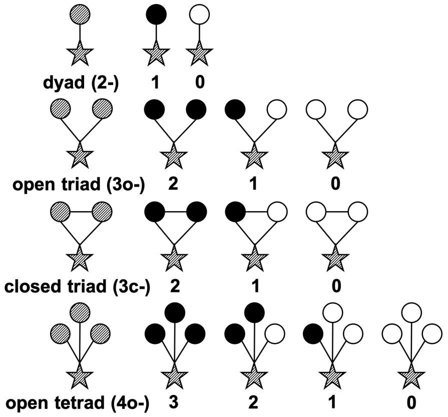

Figure 2: Examples of causal network motifs. Stars represent egos and circles represent alters. Solid indicates the node being treated, hollow indicates control, and shaded indicates that it could be treated or control. The first patterns in each row are conventional network motifs without assignment conditions, or just called *network motifs*, followed by corresponding network motifs. Our interference vector is constructed by dividing the count of a causal network motif by the count of the corresponding causal network motif. The labels below each network motif indicate the naming: for example, an open triad where one neighbor is treated is named 3o-1.

in two primary aspects. First, we focus on 1-hop ego networks that include the ego node, with the methods generalizing to higher n-hop ego networks for n>1. Second, we consider the treatment assignment conditions of the user and their n-hop connections. We use the terminology "network motifs" to refer to conventional motifs without treatment assignment labels (or assignment conditions) and causal network motifs to refer to ones with assignment conditions. Examples of network motifs are illustrated in Figure 2. We use these counts on a n-hop ego network to characterize the exposure condition of each observation.

Experimenters and practitioners need to determine a priori the region of ego networks (the n for n-hop ego networks which means the path length between an ego node and another node is no greater than n) and the network motifs that matter. For example, Figure 2 specifies 1-hop ego network and only the assignment conditions of dyads, triads, and open tetrads matter.

Using causal network motifs implies the following two assumptions:

(1) **(n-hop ego networks)** We assume that an ego node's outcome can only be affected by its own assignment condition and its (n-hop) neighbors' assignment conditions. This is a common assumption in the prior network interference literature, and is sometimes called *the stable unit treatment on neighborhood value assumption* [31, 42]. A larger n implies a more relaxed assumption.
(2) **(Specified network motifs)** Given the specified n-hop ego networks and the subgraph size for computing motifs, we assume that only the assignment conditions of specified network motifs affect the outcome of the ego observation. This assumption implies that we do not distinguish two ego networks with identical counts of specified network motifs. Considering more and higher-order would mitigate this issue.

Ideally, we should consider an infinitely large n and network motifs with more nodes because these are more relaxed assumptions. However, two issues may arise. First, it is typically computationally expensive to count network motifs of many nodes in an ego network of large n. There are many possible network motif patterns and potentially large counts. Second, related to the positivity requirement in the next section, we need all (or almost all) observations to contain all specified network motifs. Specification of too many network motifs may exclude a significant proportion of samples from the analysis.

Note that although all our examples are in the undirected setting, it can be easily extended to directed networks (e.g., a directed edge from i to j indicates that i reached j). In either undirected or directed networks, we should be cautious that the networks are pre-treatment so that the network structure is not affected by the treatment assignments. For example, when we use a network where edges represent whether the two nodes have interactions, the interactions used should have happened before the treatment was assigned; otherwise this would bring about an issue known as post-treatment bias [4].

After counting causal network motifs for each ego node in our network, our next step is to convert the counts to features which will be used in the next section. Let $\mathbf{X}_i$ denote an m-dimensional random vector, referred to as *interference vector*. The interference vector has an important requirement: Each element of the random vector is *"intervenable"* — that is, the random treatment assignment affects the value of each element of the vector. The requirement addresses the selection bias issue when we estimate the average potential outcomes.

We construct the interference vector in the following way. For each observation, for the count for each causal network motif (e.g., 2-1, 2-0, ..., 3o-2, 3o-1, ...), we normalize it by the count of the corresponding network motifs (e.g., dyads, open triads, closed triads, ...)[4]. In this way, each element of $\mathbf{X}_i$ is intervenable and the support for each element is then $[0, 1]$. Note that when considering a network motif with many nodes, some observations may not have certain network motifs, and normalization cannot be done. In these scenarios, we can either exclude this network motif from the interference vector, or drop these observations if they take a really small proportion. Please refer to Figure 3 for an illustration of constructing the interference vector.

We combine the ego node's own assignment condition Z_i and interference vector $\mathbf{X}_i$ as the features for our tree-based algorithm described in the next section to determine exposure conditions. $(Z_i, \mathbf{X}_i) \in [0, 1]^{m+1}$. Related to [8], our approach is mathematically equivalent to, $D = f(Z_i, \mathbf{X}_i)$ such that $\bar{y}(D) = \bar{y}(f(Z_i, \mathbf{X}_i))$. $\mathcal{D}$ is equivalent to partitioning $[0, 1]^{m+1}$ to $\mathcal{X}_1 \cup \mathcal{X}_2 \cup ... \cup \mathcal{X}_{|\mathcal{D}|}$. Z_i impacts direct effects, and $\mathbf{X}_i$ corresponds to indirect effects.

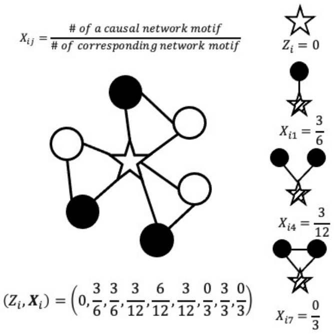
$X_{ij} = \frac{\#\ of\ a\ causal\ network\ motif}{\#\ of\ corresponding\ network\ motif}$
$Z_i = 0$
$X_{i1} = \frac{3}{6}$
$X_{i2} = \frac{3}{6}$
$X_{i4} = \frac{3}{12}$
$X_{i5} = \frac{6}{12}$
$X_{i6} = \frac{3}{12}$
$X_{i7} = \frac{0}{3}$
$X_{i8} = \frac{3}{3}$
$X_{i9} = \frac{0}{3}$
$(Z_i, \boldsymbol{X}_i) = \left(0, \frac{3}{6}, \frac{3}{6}, \frac{3}{12}, \frac{6}{12}, \frac{3}{12}, \frac{0}{3}, \frac{3}{3}, \frac{0}{3}\right)$

Figure 3: An example of ego network with treatment assignments and the corresponding interference vector. Stars represent egos and circles represent alters. Solid indicates the node being treated,hollow indicates control, and shaded indicates that it could be treated or control.

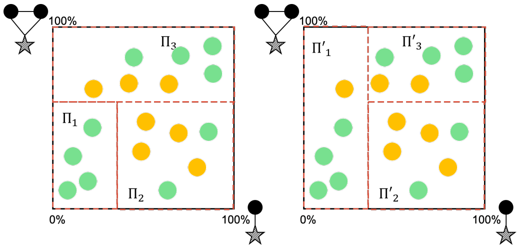

Figure 4: Illustration of selection bias and positivity. x**-axis and** y**-axis represent the fraction of the given causal network motif among the corresponding network motifs. The positions of nodes indicate observed values for each observation (not the probability distribution).** Π_1, Π_2, Π_3 **or** Π'_1, Π'_2, Π'_3 **represent a plausible partitioning. Imagine green and yellow nodes represent two types of observations (e.g. green for observations with fewer neighbors and yellow for observations with more neighbors). In independent random assignments, green nodes are more likely to have extreme values in the** x**- or** y**- axis, while yellow nodes are more likely to be centered around the mean. The left partitioning may violate positivity because yellow nodes may have zero or very small probability to belong in** Π_1**; by contrast, the right partitioning is feasible. In the right partitioning, simply taking the average is still problematic because yellow nodes have smaller probability to belong in** Π'_i**. Therefore, we need inverse probability weighting to correct this selection bias.**

5 A TREE-BASED PARTITIONING APPROACH

The next question is how to design an algorithm to determine exposure conditions by $(Z_i, \mathbf{X}_i)$ (i.e. partitioning its support $[0,1]^{m+1}$). By an abuse of notation we replace any exposure condition d by the corresponding partition $\mathcal{X}$ (where $\mathcal{X} \subset [0,1]^{m+1}$): $\pi_i(\mathcal{X}) = \mathbb{P}[(Z_i, \mathbf{X}_i) \in \mathcal{X}] = \mathbb{P}[f(Z_i, \mathbf{X}_i) = d] = \pi_i(d)$, and similarly $1_i(\mathcal{X}) = \mathbb{1}[(Z_i, \mathbf{X}_i) \in \mathcal{X}] = 1_i(d)$, or the potential outcome $y_i(\mathcal{X}) = y_i(d)$.

Our approach partitions $[0,1]^{m+1}$ and determines exposure conditions based on a decision tree regression [16].[5] Decision trees can be used for clustering [43] and typically have good interpretability [47]. Thus, it is a proper machine learning algorithm to solve the partitioning problem. Each leaf of the decision tree will determine a unique exposure condition (partition). Compared with conventional decision tree regression, we need to have the following revisions:

(1) **(Positivity)** Positivity ensures observations have a non-zero chance of being in an exposure condition [68]. In our setting, it means that for all $i \in \mathcal{U}$, $d \in \mathcal{D}$, $\pi_i(\mathcal{X}) > 0$.
If any partition d would lead to the existence of any observation i that $\pi_i(d) = 0$, we would be unable to estimate the average potential outcomes. Mathematically, it would make the denominator in Equation 3 zero. Note that the requirement is for the set all observations ($\mathcal{U}$), rather than just the observations that are randomly assigned to $\mathcal{X}$.
This is also part of the reason why we normalize the interference vector. Imagine we use the number of each causal network motif as the elements of the interference vector. If we decided a partition in which the number of 3c-1 (closed triads with fully treated neighbors) is greater than 10, then all observations with fewer than 10 open triads would have zero probability of belonging to this partition. The necessity of positivity is illustrated in Figure 4.
The tree algorithm should not split a node if such splitting will lead to any child node corresponding to a potential exposure condition in which any observation has zero probability to belong. Since $\pi_i(\mathcal{X})$ is sometimes not solvable analytically, we use Monte Carlo to approximate it ($\hat{\pi}_i(\mathcal{X})$). Moreover, we adjust the positivity requirement to non-trivial probability which allows a very few observations to have zero or near-zero probability; non-zero δ and ϵ introduce a small bias but allow partitioning more features.

$$\sum_{i \in \mathcal{U}} \mathbb{1}[\hat{\pi}_i(\mathcal{X}) \leq \epsilon] \leq \delta |\mathcal{U}|. \tag{4}$$

It means that the fraction of observations with $\hat{\pi}_i(\mathcal{X}) \leq \epsilon$ is smaller than δ.

(2) **(Honest splitting)** One pitfall is that when we estimate the variance, the algorithm is choosing a threshold to partition so that it would minimize its objective function; however, it may overfit the training data, selecting an improper threshold when splitting, and eventually overestimating the difference between the average potential outcomes indicated by the two child nodes. We thus split the original training set into training and estimation sets — the training set is used for tree partitioning and a separate estimation set is used for estimating the mean and variance. This is a common approach to correcting confidence intervals when using machine learning for causal inference [12, 41].

(3) **(Weighted sum of squared errors)** Hajek estimator can be derived by minimizing weighted sum of squared errors given a candidate exposure condition, which corresponds to a subset of $[0,1]^{m+1}$ (denoted by $\mathcal{X}$).

[4] The q-fractional neighborhood conditions are considered special cases in our approach where only dyad motifs are used.

[5] We use regressions because it also covers binary outcome variables, but our approach could be easily extended to classification.

$$\hat{y}_{Hajek}(d) = \arg\min_{y} \sum_{i \in \mathcal{U}, 1_i(\mathcal{X})=X} \frac{1}{\pi_i(\mathcal{X})}(Y_i - y)^2. \tag{5}$$

In conventional decision tree regression, one common criterion that determines whether the algorithm will continue to split a node is the reduction in sum of square errors. Since Hajek estimators can be solved through weighted linear regressions, it is sensible to use a weighted sum of squared errors (WSSE). It can be computed through a weighted linear regression where the weight is $1/\pi_i(\mathcal{X})$.

$$\text{WSSE}(\mathcal{X}) = \sum_{i \in \mathcal{U}, 1_i(\mathcal{X})=1} \frac{1}{\pi_i(\mathcal{X})}(Y_i - \hat{y}_{Hajek}(\mathcal{X}))^2. \tag{6}$$

When considering splitting the partition $\mathcal{X}$ to sub-partitions $\mathcal{X}_l$ and $\mathcal{X}_r$, we cannot simply compare $\text{WSSE}(\mathcal{X}_l)+\text{WSSE}(\mathcal{X}_r)$ versus $\text{WSSE}(\mathcal{X})$. This is because $\mathbb{E}[\sum_{i \in \mathcal{U}, 1_i(\mathcal{X})=1} \frac{1}{\pi_i(\mathcal{X})}] = \mathbb{E}[\sum_{i \in \mathcal{U}, 1_i(\mathcal{X}_l)=1} \frac{1}{\pi_i(\mathcal{X}_l)}] = \mathbb{E}[\sum_{i \in \mathcal{U}, 1_i(\mathcal{X}_r)=1} \frac{1}{\pi_i(\mathcal{X}_r)}] = |\mathcal{U}|$, $\text{WSSE}(\mathcal{X}_l) + \text{WSSE}(\mathcal{X}_r)$ is generally greater than $\text{WSSE}(\mathcal{X})$. We thus adjust the splitting criterion by taking a weighted average between $\text{WSSE}(\mathcal{X}_l)$ and $\text{WSSE}(\mathcal{X}_r)$:

$$\begin{aligned}(\mathcal{X}_l^*, \mathcal{X}_r^*) =& \arg\min_{\mathcal{X}_l, \mathcal{X}_r} \frac{\sum_i 1_i(\mathcal{X}_l)}{\sum_i 1_i(\mathcal{X})}\text{WSSE}(\mathcal{X}_l) + \\ & \frac{\sum_i 1_i(\mathcal{X}_r)}{\sum_i 1_i(\mathcal{X})}\text{WSSE}(\mathcal{X}_r), \\ \text{where } & \frac{\sum_i 1_i(\mathcal{X}_l)}{\sum_i 1_i(\mathcal{X})}\text{WSSE}(\mathcal{X}_l) + \frac{\sum_i 1_i(\mathcal{X}_r)}{\sum_i 1_i(\mathcal{X})}\text{WSSE}(\mathcal{X}_r) \\ & < \text{WSSE}(\mathcal{X}) - \gamma \\ \text{and } & \sum_i 1_i(\mathcal{X}_l) \geq \kappa \text{ and } \sum_i 1_i(\mathcal{X}_r) \geq \kappa.\end{aligned} \tag{7}$$

γ is a hyper-parameter used to require non-trivial reduction in WSSE. Similar to conventional decision trees, we can also set the minimum leaf size (κ) to prevent the tree from growing unnecessarily deep.[6] However, if the constraints cannot be satisfied, $(\mathcal{X}_l^*, \mathcal{X}_r^*)$ is an empty set and the algorithm would not further split $\mathcal{X}$.

Our algorithm is implemented by recursion (see Algorithm 1). Split is a procedure used to partition a given space. One can use $\text{Split}([0,1]^{m+1})$ to start the recursion algorithm. The data used for the algorithm is a random half of the original training set. We then use the separate half to estimate the mean and variance. Empirically, direct effects (the impact of Z_i) are usually larger than indirect effects $\mathbf{X}_i$ (the impact of $\mathbf{X}_i$). Thus, the first partition is more inclined to split on Z_i: that is, it splits into two trees, in which one corresponds to treated observations and the other corresponds to non-treated observations.

The bottleneck for computational efficiency is Line 7 — the loop that iterates every element in $\mathcal{U}$. An improvement is to replace this for-loop by randomly choosing η observations in $\mathcal{U}$. This may not help select the optimal cutoff but would provide good cutoffs if η is not too small. We also have several hyperparameters — γ, κ, δ, ϵ, and η. To tune those parameters, we can apply cross-validation to find the best choice that minimizes $\text{WSSE}^*([0,1]^{m+1})$.

Algorithm 1 Implementation for the tree-based algorithm

```
 1: procedure Split($\mathcal{X}$)
 2:     $\mathcal{X}_l^*, \mathcal{X}_r^* = \emptyset$
 3:     Compute $\hat{y}(\mathcal{X})$ and its variance using estimation set
 4:     Compute WSSE($\mathcal{X}$) using training set
 5:     $\text{WSSE}^*(\mathcal{X}) = \text{WSSE}(\mathcal{X})$
 6:     for $k \in [1, m+1]$ do
 7:         for $i \in \mathcal{U}$ do
 8:             $\theta \leftarrow \mathbf{X}_{ik}$
 9:             $\mathcal{X}_l \leftarrow \{j | \mathbf{X}_{jk} \leq \theta \ \& \ j \in \mathcal{X}\}$
10:             $\mathcal{X}_r \leftarrow \{j | \mathbf{X}_{jk} > \theta \ \& \ j \in \mathcal{X}\}$
11:             if Eq 4 is true for both $\mathcal{X}_l$ and $\mathcal{X}_r$ then
12:                 Compute WSSE($\mathcal{X}_l$), and WSSE($\mathcal{X}_r$) using training set
13:                 if Constraints in Eq. 7 are satisfied then
14:                     $\mathcal{X}_l^* \leftarrow \mathcal{X}_l$
15:                     $\mathcal{X}_r^* \leftarrow \mathcal{X}_r$
16:                     Update $\text{WSSE}^*(\mathcal{X})$ using Equation 7
17:                 end if
18:             end if
19:         end for
20:     end for
21:     if $\mathcal{X}_l^* \neq \emptyset$ and $\mathcal{X}_r^* \neq \emptyset$ then ▷ If find any satisfy Eq. 7, choose the
    one with minimum WSSE.
22:         Split($\mathcal{X}_l^*$)
23:         Split($\mathcal{X}_r^*$)
24:     else
25:         Add new exposure condition to $\mathcal{D}$, corresponding to $\mathcal{X}$
26:     end if
27: end procedure
```

In addition to estimating the average potential outcome, our approach can also be applied to estimating global average treatment effects and heterogeneous direct effects with indirect effects fixed as explained:

(1) **Global average treatment effects.** The global average treatment effect of an experiment is the difference between the average outcome in a counterfactual world where everyone is treated versus that in a counterfactual world where everyone is non-treated [63]. This is fundamentally unsolvable unless assumptions such as SUTVA or our assumptions are made. In our approach, these two counterfactual worlds belong to two separate partitions. We can just identify these two partitions and compare the difference in the two average potential outcomes to compute the global average treatment effect.
When the tree algorithm terminates, we have two special subsets of $[0,1]^{m+1}$, denoted by $\mathcal{X}_0$ and $\mathcal{X}_1$ respectively. $\mathcal{X}_0$ is the partition that contains the cases where $Z_i = 0$, all elements of $\mathbf{X}_i$ that represent fully non-treated neighborhood (2-0, 3o-0, 3c-0, 4o-0, ...) are equal to 1, and the rest equal 0; $\mathcal{X}_1$ is the partition that contains the cases where $Z_i = 1$ and all elements of $\mathbf{X}_i$ that represent fully treated neighborhood (2-1, 3o-2, 3c-2, 4o-2, ...) are equal to 1, and the rest equal 0. Then we can use $\hat{y}(\mathcal{X}_1) - \hat{y}(\mathcal{X}_0)$ to estimate the global

[6] An empirically effective alternative is to require $\sum_{i \in \mathcal{U}, 1_i(\mathcal{X})=1} \frac{1}{\hat{\pi}_i(\mathcal{X})}$ to be very close to $|\mathcal{U}|$; e.g., $(1-\varepsilon)|\mathcal{U}| \leq \sum_{i \in \mathcal{U}, 1_i(\mathcal{X})=1} \frac{1}{\hat{\pi}_i(\mathcal{X})} \leq (1+\varepsilon)|\mathcal{U}|$, where ε is a small number. This helps us to avoid partitions with a small number of observations, which may have a high degree of randomness in $\sum_{i \in \mathcal{U}, 1_i(\mathcal{X})=1} \frac{1}{\hat{\pi}_i(\mathcal{X})}$ and consequently, $\hat{y}_{Hajek}(\mathcal{X})$.

treatment effect. Variance can be estimated by the method proposed in [8].

(2) **Heterogeneous direct effects with indirect effects fixed.** As discussed, our algorithm tends to split on Z_i first if direct effects are larger than indirect effects. Once the tree grows to two sub-trees that correspond to treated and non-treated observations, respectively, the further partitioning is not synchronized in these two trees. However, what if we want to understand how direct effects vary across different network interference conditions? Let $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(0)}$ be the two spaces such that $Z_i^{(1)} = 1$ for all $(Z_i^{(1)}, \mathbf{X}_i^{(1)}) \in \mathcal{X}^{(1)}$; and $Z_i^{(0)} = 0$ for all $(Z_i^{(0)}, \mathbf{X}_i^{(0)}) \in \mathcal{X}^{(0)}$. $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(0)}$ are complementary if for all $(0, \mathbf{X}_i^{(0)}) \in \mathcal{X}_i^{(0)}$, we have $(1, \mathbf{X}_i^{(0)}) \in \mathcal{X}_i^{(1)}$, and for all $(1, \mathbf{X}_i^{(1)}) \in \mathcal{X}_i^{(1)}$, we have $(0, \mathbf{X}_i^{(1)}) \in \mathcal{X}_i^{(0)}$.
We can then revise our tree algorithm by only partitioning on a space $[0, 1]^m$ (i.e., on $\mathbf{X}_i$) and not partitioning on Z_i. In each node, including leaves, the goal is no longer estimating the potential outcome. Instead, we estimate the heterogeneous direct effects ($\tilde{\mathcal{X}} = \{\mathbf{X} | (1, \mathbf{X}) \in \mathcal{X}^{(1)}\}$)

$$\tau_{direct}(\tilde{\mathcal{X}}) = \bar{y}_{Hajek}(\mathcal{X}^{(1)}) - \bar{y}_{Hajek}(\mathcal{X}^{(0)}) \tag{8}$$

$\mathcal{X}^{(1)}$ indicates the cases where neighborhood assignment conditions are in a given region (e.g., high structural diversity, high echo chamber, or any other interference conditions) and the ego is treated; and $\mathcal{X}^{(0)}$ indicates the cases where the ego is non-treated but neighborhood interference conditions are the same. In this case, $\tau_{direct}(\tilde{\mathcal{X}})$ represents the average direct effect under that interference conditions.
If estimating heterogeneous direct effects with indirect effects fixed, two main revisions in Algorithm 1 are made. First, we adjust the weighted linear regressions. Remember that the coefficient for the constant variable represented the Hajek estimator; in this case, we need to add the Z_i into the regression, and report the coefficient for this variable to estimate $\tau_{direct}(\tilde{\mathcal{X}})$. WSSE is still derived from the error term for the weighted linear regression. Second, the variance of the estimator is more complicated to estimate. Essentially, the variance estimation is similar to the methods proposed in [12].

6 EXPERIMENTS

We evaluate our algorithm in a synthetic Watts-Strogatz network where we can verify our approach recovers the ground-truth which we know, and a real-world A/B test of a new product feature on Facebook to illustrate the scalability of our approach.

Watts-Strogatz Simulation Network. In studies of causal inference, a common challenge is that the ground-truth potential outcomes under both treatment and control for a given unit i is missing — that is, for each observation, we can only observe one single potential outcome given either a treatment or control assignment. Therefore, we rely on a simulation study which is often used to verify the effectiveness of causal inference methods.

We generate a Watts-Strogatz network [67] with $|\mathcal{U}|$=200,000. The Watts-Strogatz model is a random graph generator that preserves network properties such as clustering and the "small-world phenomenon". To ensure a large variation in the local structure of individuals' neighborhood, we have a high rewiring rate for edges — 50%.

To be as general as possible and also complement the real-world independent assignment experiment, we use graph cluster randomization [63] in the simulation study. We use a simple clustering approach — we cluster every 10 nodes on the ring of the WS network, and we assign the treatment randomly and independently on the cluster level. We consider the following four different data-generating processes for outcomes, assuming that ε_i is randomly drawn from Gaussian distribution with the mean of 0 and the variance of 1:

(1) **Cutoff Outcome**.

$$Y_i^{(1)} = 0.1|\mathcal{N}_i| + gender_i + 2Z_i \times \mathbb{1}[X_{i,3c-2} > 0.7] + \varepsilon_i. \tag{9}$$

$gender_i$ is a covariate independent of any other variables, which is randomly assigned to 1 or 0. Since this function has clear cutoffs, we can use it to validate that our tree-based algorithm splits on the corresponding cutoff.

(2) **Causal Structural Diversity Outcome**.

$$\begin{aligned} Y_i^{(2)} = 0.1|\mathcal{N}_i| + gender_i + structural\_diversity\_of\_treated + \\ Z_i \times structural\_diversity\_of\_treated + \varepsilon_i \end{aligned} \tag{10}$$

Structural diversity is the number of disjointed components in an observation's neighborhood. "*structural_diversity_of_treated*" is the structural diversity given the set of treated neighbors. Although it was found that structural diversity predicts a higher product adoption rate [62], it is unclear whether this is a causal impact or just correlation. A causal impact means it is actually the structural diversity for the treated neighbors that matters, while correlation means that the behavior is reflected by network structure, regardless of assignment conditions [60]. Our approach can analyze whether the causal impact of structural diversity exists in experimental data. Especially, the numbers of fully treated open triads or tetrads indicate the structure diversity of treated neighbors (3c-2 or 4o-3), and we expect our algorithm to split on these features.

(3) **Correlational Structural Diversity Outcome**.

$$\begin{aligned} Y_i^{(3)} = 0.1|\mathcal{N}_i| + gender_i + structural\_diversity + \\ Z_i \times structural\_diversity + \varepsilon_i \end{aligned} \tag{11}$$

"*structural_diversity*" is the structural diversity given all neighbors, regardless of their assignment conditions. Therefore, we do not expect the tree algorithm to split on any features.

(4) **Validation Under Null Effect**.

$$Y_i^{(4)} = |\mathcal{N}_i| + \varepsilon_i. \tag{12}$$

As a sanity check, we use a covariate — the number of neighbors as the outcome variable. We do not expect the tree algorithm to split on any features.

Results are presented in Figure 5 and summarized below:

(1) **Cutoff Outcome** ($Y_i^{(1)}$): As expected, after splitting treated versus control, it splits on fully treated closed triads (3c-2), with a threshold of 69%. This is consistent with the parameter 0.7 set in Equation 9. Also, as expected, it does not further split on control groups. However, the algorithm has an unexpected split on 3o-1, although the difference between the resulting d_1 and d_2 is not statistically significant. With further investigation, we find that this is because when partitioning the tree using the training set, it overfits the noise; however, since we use the estimation set for the average potential outcome and its variance (i.e. honest splitting), the resulting tree does not show this significant difference. This result also demonstrates the importance of honest splitting.

(2) **Causal Structural Diversity Outcome** ($Y^{(2)}$): The result is presented in the right panel in Figure 5. It first splits on treated versus control. For treated observations, it first splits on fully treated open tetrad (4o-3), which is positively correlated with the degree of structural diversity of treated neighbors. When the fraction of fully treated open tetrad is greater than 6.6%, the algorithm terminates splitting, giving a partition (exposure condition) with the largest average potential outcome. If the fraction of fully treated open tetrad is smaller than 6.6%, it further splits on open triads (3o-1). If the fraction of 3o-1 is greater than 40%, the resulting exposure condition gives the smaller average potential outcome within treated observations. For control observations, partially because the structural diversity plays a smaller effect in the outcome function, it simply splits on the dyad level and terminates.

(3) **Correlational Structural Diversity Outcome** ($Y^{(3)}$) and **Validation Under Null Effect** ($Y^{(4)}$): As expected, it terminates after splitting samples into treated and control groups. Note these examples have nothing to illustrate.

We compare our results with a baseline method used in [8]. The paper provides an examine of four exposure conditions: no effect, direct effect, indirect effect, and direct + indirect effects. For example, when $Y_i^{(1)}$ is the outcome, most observations belong to either "indirect effect" (1.4981 ± 0.0035, 48.1%) or "direct + indirect effects" (2.9182 ± 0.0034, 50.0%). Similar results are derived in other outcomes. Therefore, this specification approach is not suitable for our experimental data. Moreover, such a partition cannot reveal the effects of specific network motifs or understand theories such as structural diversity.

We also compare our result the specification of the fractional q-neighborhood exposure condition in [63], which is equivalent to the proposed algorithm using dyad features only. This comparison can help highlight the importance of accounting for network motifs rather than simply counting treated friends. . When using only dyad features, the algorithm splits treated observations on 2-1 (i.e. fraction of treated neighbors), resulting in an average of potential outcome of 2.0233 (±0.0192) and 3.2133 (±0.0084) for less than or greater than 60%, respectively. It does not split the control observations either.

We use estimated global treatment effects [23, 63] as a reference for the advantage of using causal network motif over simply accounting for proportions of treatment neighbors. Here we use $Y_i^{(1)}$ but similar results are derived in other settings. Using dyads features only (i.e., the proportion of treated neighbors, see Figure 6) gives (3.2137 − 1.4976 =) 1.7161 (±0.0112) while using all proposed motif features give (3.5038 − 1.5057 =) 1.9981 (±0.0102); as a baseline, the true global treatment effect is 2.0000, and the average treatment effect under SUTVA is (2.9262 − 1.5057 =) 1.4205 (±0.0118). In sum, using network motifs helps reduce more bias than using dyads features only when we estimate the global treatment effects because it characterizes network complex structure rather than purely counting the fraction of treated neighbors.

We also investigate heterogeneous direct effects given different network interference conditions. As shown in Figure 7, for both $Y^{(1)}$ and $Y^{(2)}$, it splits on the correct threshold: for $Y^{(1)}$, Z_i only matters then the fraction of 3c-2 is greater than 69%. For $Y^{(2)}$, it splits on important feature (4o-3), and the heterogeneous direct effect is larger when this fraction is greater.[7] As expected, no effect is observed for $Y^{(3)}$ and $Y^{(4)}$.

Care Reaction Rollout. Finally, we apply our approach on a real-world A/B test of a product launch at Facebook. Facebook launched the 7th reaction, "Care".[8] When the reaction was launched, users in the control group (50%) could not use the Care reaction but could still observe other users reacting with care. The treated group (the other (50%), could both use and see the Care reaction. The sample size is approximately 5% of the Facebook population, and we only take into account users and neighbors who are in this 5% sampled users. Note that we cannot compare treated users in this 5% versus all the rest of Facebook population because the database does not log non-compliance: those who were assigned to the 5% rollout but did not activate the treatment assignment (non-active users who did not attempt to use Care) are not recorded in the database and thus such a comparison is biased by non-compliance. Another reason is that a 50:50 test/control ratio would provide more variation in the proportion of each causal network motif.

In this experiment, we expect a user's use of reactions to be impacted by the number of their Facebook friends who could use Care and how those friends are connected which might impact their usage with each other. If a user has more friends using Care, or a group of friends able to use Care, she might use more Care reactions. There are two main outcome variables in this A/B test: (1) the number of Care reaction uses during the week after the experiment was launched. In the control group, the number of Care reaction uses is always zero; and (2) the number of other reaction uses. To prevent the impact from extreme points, outcome variables are set on the log scale ($\log_{10}(x+1)$).

We randomly split our observations into approximately equal training and estimation sets of sufficiently large sample sizes. We compute only dyad and triad network motifs, which is mainly restricted by computational resources.

[7] For the tasks of heterogeneous direct effects with indirect effects fixed, the average direct effect for a parent node does not necessarily lies between the values of its two child nodes.

[8] https://www.facebook.com/careers/life/the-story-of-facebooks-care-reaction

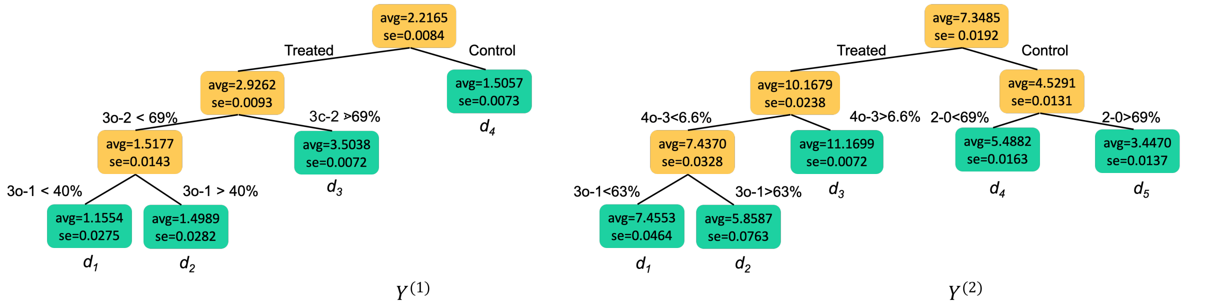

Figure 5: The result trees for the simulation experiment using all specified network motifs. The two trees represent $Y^{(1)}$ or the Validation Cutoff Outcome (left) and $Y^{(2)}$ or the Causal Structural Diversity Outcome (right), respectively. The numbers in each leaf represents the average potential outcome and standard error (square root of variance) of the corresponding partition (exposure condition).

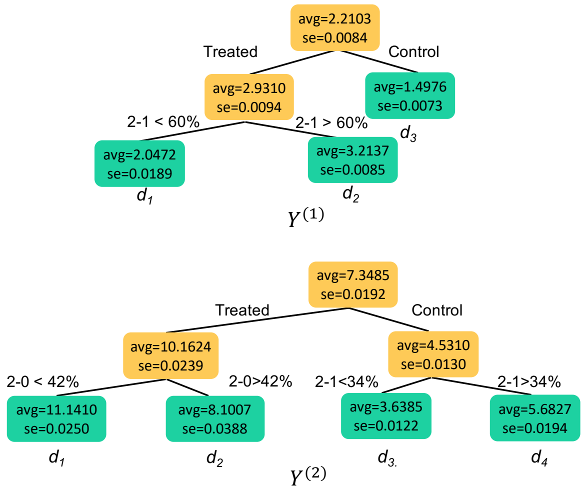

Figure 6: The result trees for the simulation experiment using dyad network motifs only. The two trees represent $Y^{(1)}$ or the Cutoff Outcome (top) and $Y^{(2)}$ or the Causal Structural Diversity Outcome (bottom), respectively. The numbers in each leaf represents the average potential outcome and standard error (square root of variance) of the corresponding partition (exposure condition).

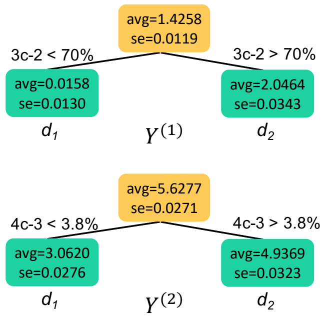

Figure 7: The result trees for heterogeneous direct effects for the simulation experiment using all specified network motifs. The numbers in each node or leaf represents the average direct effect given the network interference condition and standard error (square root of variance) of the corresponding partition (exposure condition).

The results are presented in Figure 8. There are 8 exposure conditions determined by the algorithm when the outcome is other reactions, and 4 when outcome is use of Care. Again, we observe heterogeneity among different network interference conditions. For example, on the right panel, d_1 and d_2 have a difference of $(1.1601 - 1.1365 =)\ 0.0236$. As a reference, directly comparing the averages between treated and control groups gives -0.0293, which shows the importance of distinguishing between different network interference conditions.

Again, we also compare the results using all specified network motifs (Figure 8) versus the results using only dyads (Figure 9) to the importance of using network motifs beyond dyads. In the left panel of Figure 8, d_3 contains the scenario of a fully treated neighborhood and d_5 the scenario of a fully non-treated neighborhood. Therefore, we can estimate the global average treatment as -0.0394 (± 0.0049). The baseline average treatment effect by directly comparing the averages between treated and control groups is -0.0293 (± 0.0021). Thus, simply comparing the difference in means between treated and control groups could underestimate the true treatment effect by 25%. By contrast, in the left panel of Figure 9, comparing d_2 and d_3 only gives $(1.1516 - 1.1690 =)\ -0.0174$ (± 0.0022). Similar analysis and conclusions also apply when the outcome is Care.

We also examine heterogeneous direct effects under different network interference conditions. Since use of Care is always zero for the control group, this analysis is only meaningful when the outcome is other reactions. The algorithm splits on 3o-2 at 23% but does not provide significant differences between the two child nodes.

7 DISCUSSION

Network interference is much more complicated than simply being described as the indirect effect. To examine and analyze heterogeneity of the causal impact of neighbors in experimental datasets, we provide a two-step solution. We first employ causal network motifs to characterize the network interference conditions, and then develop a tree-based algorithm for partitioning. Our tree-based algorithm is interpretable in terms of highlighting which exposure

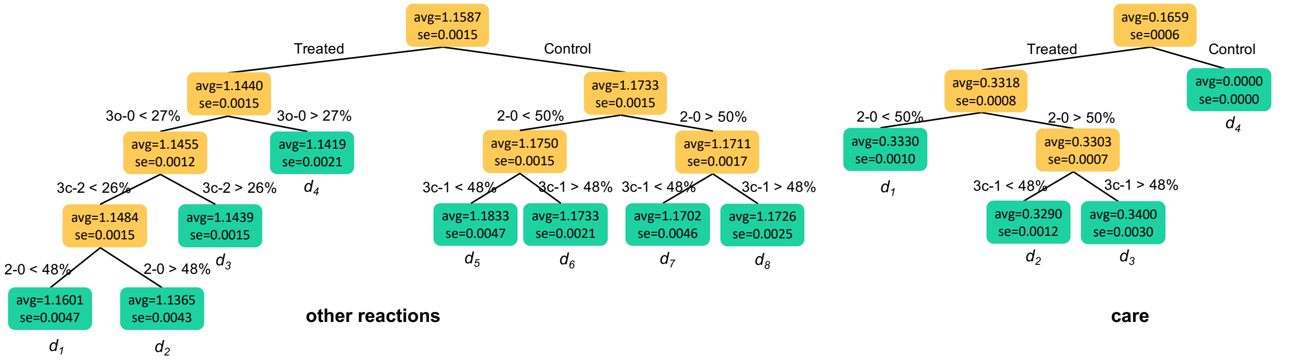

Figure 8: The result trees for the Care experiment using all specified network motifs. The left panel presents the result for the use of other reactions, and the right panel presents the result for the use of Care. The numbers in each leaf represents the average potential outcome and standard error (square root of variance) of the corresponding partition (exposure condition).

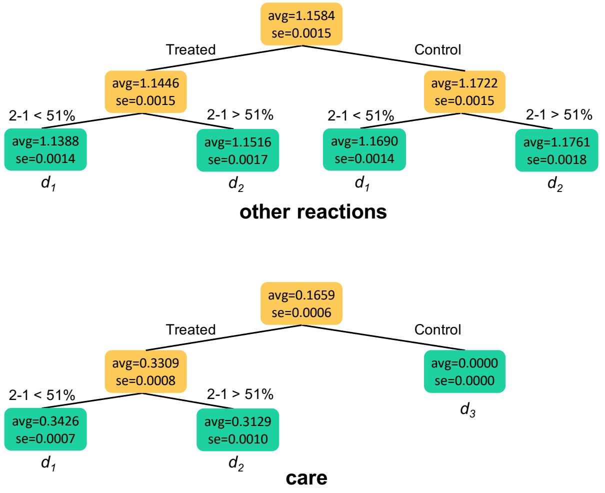

Figure 9: The result trees for the Care experiment using dyads only. The left panel presents the result for the use of other reactions, and the right panel presents the result for the use of Care. The numbers in each leaf represents the average potential outcome and standard error (square root of variance) of the corresponding partition.

conditions matter for defining potential outcomes, addresses selection bias and positivity issues, and avoids incorrect standard error concerns via honest splitting.

Given the large number of researcher degrees of freedom in existing approaches for network interference such as choosing the threshold for an exposure condition, our approach provides a simple way to automatically specify exposure conditions. In this way, researchers no longer need to define exposure conditions a priori, and the exposure conditions generated by the algorithm are suitable for the given data and experiment. We believe that methodological innovation for addressing network interference concerns in A/B tests on networks will continue to be an important area for development, and accounting for network motifs with treatment assignment conditions provides a useful way to detect heterogeneous network interference effects.

We also highlight our approach should not be confused with the Causal Tree and relevant methods [12, 66]. The goal of our approach is to partition on the space of random assignments, while the Causal Tree and similar methods partition only on the covariate space. The critical difference in our setting is that in each partition of the treatment space, all observations should have a (almost) non-zero probability of belonging to that partition. In this way, we can construct certain unbiased estimators, such as Hajek, to estimate the average outcome under certain treatment regions and thus quantify a causal impact. By contrast, in Causal Tree and relevant methods, covariates are not intervened by the experiment and each observation has only probability of zero or one of belonging to each partition. Therefore, these methods are only used to identify heterogeneous treatment effects in subsets of exposure conditions determined by covariates, but not for the causal impact of heterogeneous network interference conditions for all population.

Practitioners using our approach may obtain important insights. For example, they could understand how to utilize social contagion for product promotion when they have constraints on the number of promos. Researchers may identify important network interference conditions that are not theorized in certain experimental settings.

There remain many open questions directions based on our approach. First, we can incorporate covariates into the algorithm such as demographic features. One way is to also allow partitioning on the covariate space as well as the treatment space as in this work. However, once the algorithm splits on a covariate, all the descendants of that node only estimate the average potential outcome for the subsample that satisfies the criterion. The other way is to incorporate covariates in the weighted linear regressions in our algorithm. This helps reduce the variance for the estimators and improve the precision when estimating average potential outcomes. We may also want to account for tie strength as well.

Second, we may consider alternative algorithms. Decision trees are not the only choice in our setting. We use a decision tree partially because it is an interpretable machine learning algorithm and does not involve functional form specification, except for assuming constant potential outcome. Our tree-based algorithm can in fact be improved by the Hoeffding tree [25], which provides a streaming algorithm to perform the partitioning efficiently. Instead of using sample split, we may improve our methods by the conformal prediction theory [58]. Moreover, we can imagine using nearest neighbor based algorithms and local regression instead to estimate the potential outcome given any point or region in $[0,1]^{m+1}$. In addition, parametric methods can also be used when the goal is specific about estimating the potential outcome at certain points (e.g., fully treated neighbors) or estimating the global treatment effect [23].

Finally, our approach can be extended to any experimental data with multiple variables or continuous treatment variables, which are not necessarily experiments on social networks. While existing causal inference literature has primarily studied single binary or categorical treatments, fewer studies have approached continuous or multiple variables [35, 38]. Our approach provides a way to automatically convert multiple or continuous treatment variables to categorical treatments. It would be interesting to further investigate how machine learning can be applied to causal inference for continuous or multiple treatments as well as adapting this approach to observational causal inference settings.

Code Availability Statement: https://github.com/facebookresearch/CausalMotifs

Acknowledgements: We thank Lada Adamic, Dean Eckles, Brian Karrer, and Steven Li for their helpful comments. We also thank “Fresh from the arXiv” seminar participants at the University of Oxford and the MIT Conference on Digital Experimentation (CODE) participants for useful comments and feedback.

REFERENCES

[1] Christoph Adami, Jifeng Qian, Matthew Rupp, and Arend Hintze. 2011. Information content of colored motifs in complex networks. *Artificial Life* 17, 4 (2011), 375–390.

[2] Nesreen K Ahmed, Jennifer Neville, Ryan A Rossi, Nick G Duffield, and Theodore L Willke. 2017. Graphlet decomposition: Framework, algorithms, and applications. *Knowledge and Information Systems* 50, 3 (2017), 689–722.

[3] Uri Alon. 2007. Network motifs: theory and experimental approaches. *Nat Rev Genet* 8, 6 (2007), 450–461.

[4] Joshua D Angrist and Jörn-Steffen Pischke. 2008. *Mostly harmless econometrics: An empiricist's companion.* Princeton University Press.

[5] Elliott M Antman, Joseph Lau, Bruce Kupelnick, Frederick Mosteller, and Thomas C Chalmers. 1992. A comparison of results of meta-analyses of randomized control trials and recommendations of clinical experts: treatments for myocardial infarction. *JAMA* 268, 2 (1992), 240–248.

[6] Sinan Aral. 2016. Networked experiments. *The Oxford Handbook of the Economics of Networks* (2016), 376–411.

[7] Sinan Aral and Dylan Walker. 2011. Creating social contagion through viral product design: A randomized trial of peer influence in networks. *Manage Sci* 57, 9 (2011), 1623–1639.

[8] Peter M Aronow and Cyrus Samii. 2017. Estimating average causal effects under general interference, with application to a social network experiment. *Ann Appl Stat* (2017).

[9] Bruno Arpino, Luca De Benedictis, and Alessandra Mattei. 2015. Implementing propensity score matching with network data: The effect of GATT on bilateral trade. (2015).

[10] Susan Athey. 2015. Machine learning and causal inference for policy evaluation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 5–6.

[11] Susan Athey, Dean Eckles, and Guido W Imbens. 2018. Exact p-values for network interference. *JASA* 113, 521 (2018), 230–240.

[12] Susan Athey and Guido Imbens. 2016. Recursive partitioning for heterogeneous causal effects. *PNAS* (2016).

[13] Eytan Bakshy, Dean Eckles, Rong Yan, and Itamar Rosenn. 2012. Social influence in social advertising: evidence from field experiments. In *EC.* 146–161.

[14] Eytan Bakshy, Solomon Messing, and Lada A Adamic. 2015. Exposure to ideologically diverse news and opinion on Facebook. *Science* 348, 6239 (2015), 1130–1132.

[15] Guillaume Basse and Avi Feller. 2018. Analyzing two-stage experiments in the presence of interference. *JASA* 113, 521 (2018), 41–55.

[16] Christopher M Bishop. 2006. *Pattern Recognition and Machine Learning.*

[17] Jake Bowers, Mark M Fredrickson, and Costas Panagopoulos. 2013. Reasoning about interference between units: A general framework. *Polit Anal* (2013), 97–124.

[18] Damon Centola. 2010. The spread of behavior in an online social network experiment. *Science* (2010).

[19] Damon Centola and Michael Macy. 2007. Complex contagions and the weakness of long ties. *AJS* 113, 3 (2007), 702–734.

[20] Deepayan Chakrabarti and Christos Faloutsos. 2006. Graph mining: Laws, generators, and algorithms. *ACM computing surveys (CSUR)* 38, 1 (2006).

[21] Jin Chen, Wynne Hsu, Mong Li Lee, and See-Kiong Ng. 2007. Labeling network motifs in protein interactomes for protein function prediction. In *ICDM.* 546–555.

[22] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, and Whitney Newey. 2017. Double/debiased/neyman machine learning of treatment effects. *Am Econ Rev* (2017).

[23] Alex Chin. 2019. Regression adjustments for estimating the global treatment effect in experiments with interference. *JCI* 7, 2 (2019).

[24] Diane J Cook and Lawrence B Holder. 2006. *Mining graph data.* John Wiley & Sons.

[25] Pedro Domingos and Geoff Hulten. 2000. Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining.* 71–80.

[26] Dean Eckles and Eytan Bakshy. 2020. Bias and high-dimensional adjustment in observational studies of peer effects. *JASA* (2020).

[27] Dean Eckles, Brian Karrer, and Johan Ugander. 2016. Design and analysis of experiments in networks: Reducing bias from interference. *JCI* 5, 1 (2016).

[28] Naoki Egami, Christian J Fong, Justin Grimmer, Margaret E Roberts, and Brandon M Stewart. 2018. How to make causal inferences using texts. *arXiv preprint* (2018).

[29] RA Fisher. 1937. *The design of experiments.* Number 2nd Ed. Oliver & Boyd, Edinburgh & London.

[30] Seth Flaxman, Sharad Goel, and Justin M Rao. 2016. Filter bubbles, echo chambers, and online news consumption. *Public Opin Q* 80, S1 (2016), 298–320.

[31] Laura Forastiere, Edoardo M Airoldi, and Fabrizia Mealli. 2020. Identification and estimation of treatment and interference effects in observational studies on networks. *JASA* (2020), 1–18.

[32] Brian Gallagher and Tina Eliassi-Rad. 2008. Leveraging label-independent features for classification in sparsely labeled networks: An empirical study. In *International Workshop on Social Network Mining and Analysis.* Springer, 1–19.

[33] Alan S Gerber and Donald P Green. 2012. *Field experiments: Design, analysis, and interpretation.* WW Norton.

[34] Cassandra Handan-Nader, Daniel E Ho, and Becky Elias. 2020. Feasible Policy Evaluation by Design: A Randomized Synthetic Stepped-Wedge Trial of Mandated Disclosure in King County. *Eval Rev* (2020).

[35] Keisuke Hirano and Guido W Imbens. 2004. *Applied Bayesian modeling and causal inference from incomplete-data perspectives.* Vol. 226164. Chapter The propensity score with continuous treatments, 73–84.

[36] Daniel E Ho. 2017. Does peer review work: An experiment of experimentalism. *Stan L Rev* 69 (2017), 1.

[37] Kosuke Imai, Zhichao Jiang, and Anup Malani. 2020. Causal inference with interference and noncompliance in two-stage randomized experiments. *JASA* (2020), 1–13.

[38] Kosuke Imai and David A Van Dyk. 2004. Causal inference with general treatment regimes: Generalizing the propensity score. *JASA* 99, 467 (2004), 854–866.

[39] Guido W Imbens and Donald B Rubin. 2010. Rubin Causal Model. In *Microeconometrics.* Springer, 229–241.

[40] Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. 2013. Online controlled experiments at large scale. In *KDD.* 1168–1176.

[41] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. 2019. Metalearners for estimating heterogeneous treatment effects using machine learning. *PNAS* 116, 10 (2019), 4156–4165.

[42] Michael P Leung. 2020. Treatment and spillover effects under network interference. *Rev Econ Stat* 102, 2 (2020), 368–380.

[43] Bing Liu, Yiyuan Xia, and Philip S Yu. 2000. Clustering through decision tree construction. In *CIKM.* 20–29.

[44] Anne B Loucks and Jean R Thuma. 2003. Luteinizing hormone pulsatility is disrupted at a threshold of energy availability in regularly menstruating women. *J Clin Endocrinol Metab* 88, 1 (2003), 297–311.

[45] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. 2002. Network motifs: simple building blocks of complex networks. *Science* 298, 5594 (2002), 824–827.

[46] Jean Pouget-Abadie, Guillaume Saint-Jacques, Martin Saveski, Weitao Duan, S Ghosh, Y Xu, and Edoardo M Airoldi. 2019. Testing for arbitrary interference on experimentation platforms. *Biometrika* 106, 4 (2019), 929–940.

[47] J. Ross Quinlan. 1986. Induction of decision trees. *Mach Learn* (1986).

[48] Pedro Ribeiro and Fernando Silva. 2014. Discovering colored network motifs. In *Complex Networks.* Springer, 107–118.

[49] Margaret E Roberts, Brandon M Stewart, and Richard A Nielsen. 2018. Adjusting for confounding with text matching. *AJPS* (2018).

[50] Paul R Rosenbaum. 2007. Interference between units in randomized experiments. *JASA* 102, 477 (2007), 191–200.

[51] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (1983), 41–55.
[52] Donald B Rubin. 2005. Causal inference using potential outcomes: Design, modeling, decisions. *JASA* 100, 469 (2005), 322–331.
[53] Anida Sarajlić, Noël Malod-Dognin, Ömer Nebil Yaveroğlu, and Nataša Pržulj. 2016. Graphlet-based characterization of directed networks. *Sci Rep* 6 (2016), 35098.
[54] CE Särndal, B Swensson, and J Wretman. 1992. *Model assisted survey sampling Springer*. Springer.
[55] Martin Saveski, Jean Pouget-Abadie, Guillaume Saint-Jacques, Weitao Duan, Souvik Ghosh, Ya Xu, and Edoardo M Airoldi. 2017. Detecting network effects: Randomizing over randomized experiments. In *KDD*.
[56] Fredrik Sävje, Michael J Higgins, and Jasjeet S Sekhon. 2017. Generalized full matching and extrapolation of the results from a large-scale voter mobilization experiment. *arXiv preprint* (2017).
[57] Shaun R Seaman and Ian R White. 2013. Review of inverse probability weighting for dealing with missing data. *Stat Methods Med Res* (2013).
[58] Glenn Shafer and Vladimir Vovk. 2008. A Tutorial on Conformal Prediction. *Journal of Machine Learning Research* 9, 3 (2008).
[59] Jerzy Splawa-Neyman, Dorota M Dabrowska, and TP Speed. 1990. On the application of probability theory to agricultural experiments. *Stat Sci* (1990), 465–472.
[60] Jessica Su, Krishna Kamath, Aneesh Sharma, Johan Ugander, and Sharad Goel. 2020. An Experimental Study of Structural Diversity in Social Networks. In *ICWSM*, Vol. 14. 661–670.
[61] Eric J Tchetgen Tchetgen and Tyler J VanderWeele. 2012. On causal inference in the presence of interference. *Stat Methods Med Res* (2012).
[62] Johan Ugander, Lars Backstrom, Cameron Marlow, and Jon Kleinberg. 2012. Structural diversity in social contagion. *PNAS* 109, 16 (2012), 5962–5966.
[63] Johan Ugander, Brian Karrer, Lars Backstrom, and Jon Kleinberg. 2013. Graph cluster randomization: Network exposure to multiple universes. In *KDD*. 329–337.
[64] Johan Ugander and Hao Yin. 2020. Randomized Graph Cluster Randomization. *arXiv preprint arXiv:2009.02297* (2020).
[65] Tyler J VanderWeele. 2008. Ignorability and stability assumptions in neighborhood effects research. *Stat Med* (2008).
[66] Stefan Wager and Susan Athey. 2018. Estimation and inference of heterogeneous treatment effects using random forests. *JASA* (2018).
[67] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world'networks. *Nature* 393, 6684 (1998), 440–442.
[68] Daniel Westreich and Stephen R Cole. 2010. Invited commentary: positivity in practice. *Am J Epidemiol* (2010).
[69] Jeffrey C Wong. 2020. Computational Causal Inference. *arXiv preprint* (2020).
[70] Ya Xu, Nanyu Chen, Addrian Fernandez, Omar Sinno, and Anmol Bhasin. 2015. From infrastructure to culture: A/B testing challenges in large scale social networks. In *KDD*. 2227–2236.