Almost-Matching-Exactly for Treatment Effect Estimation under Network Interference

M. Usaid Awan **Marco Morucci** **Vittorio Orlandi**
Sudeepa Roy **Cynthia Rudin** **Alexander Volfovsky**
Duke University

Abstract

We propose a matching method that recovers direct treatment effects from randomized experiments where units are connected in an observed network, and units that share edges can potentially influence each others' outcomes. Traditional treatment effect estimators for randomized experiments are biased and error prone in this setting. Our method matches units almost exactly on counts of unique subgraphs within their neighborhood graphs. The matches that we construct are interpretable and high-quality. Our method can be extended easily to accommodate additional unit-level covariate information. We show empirically that our method performs better than other existing methodologies for this problem, while producing meaningful, interpretable results.

1 INTRODUCTION

Randomized experiments are considered to be the gold standard for estimating causal effects of a treatment on an outcome. Typically, in these experiments, the outcome of a unit is assumed to be only affected by the unit's own treatment status, and not by the treatment assignment of other units (Cox, 1958; Rubin, 1980). However, in many applications – such as measuring effectiveness of an advertisement campaign or a teacher training program – units interact, and ignoring these interactions results in poor causal estimates (Halloran and Struchiner, 1995; Sobel, 2006). We propose a method that leverages the observed network structure of interactions between units to account for treatment interference among them.

We study a setting in which a treatment has been uniformly randomized over a set of units connected in a network, and where treatments of connected units can influence each others' outcomes. The development of methods for this setting is a relatively new field in causal inference methodology, and only few approaches for it have been proposed (e.g., van der Laan, 2014; Aronow et al., 2017; Sussman and Airoldi, 2017).

In this paper, we propose a method that leverages matching (Rosenbaum and Rubin, 1983) to recover direct treatment effects from experiments with interference. Our method makes several key contributions to the study of this setting: First, *our method explicitly leverages information about the network structure of the experimental sample to adjust for possible interference while estimating direct treatment effects.* Second, unlike other methods, matching allows us to *nonparametrically* estimate treatment effects, without the need to specify parametric models for interference or outcomes. Third, matching produces highly interpretable results, informing analysts as to which features of the input data were used to produce estimates. More specifically, *we match on features of graphs that are easy to interpret and visualize.*

In our setting, units experience interference according to their neighborhood graphs – the graphs defined by the units they are directly connected to. Units with similar neighborhood graphs will experience similar interference. For example, the educational outcome of a student randomly assigned to an extra class depends on whether or not her friends are also assigned to that class, and not just on how many: the specific structure of the student's friendship circle will influence whether or not study groups are formed, how information is shared, how much attention the treated student will devote to the class, and so on. All of this will impact the overall educational outcomes of interest.

Because of this, matching units with similar neighborhood graphs together will enable us to recover direct treatment effects even under interference. We match units' neighborhood graphs on counts of sub-

Proceedings of the 23$^{\text{rd}}$International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

graphs within them, as graphs with similar counts of the same unique subgraphs are naturally likely to be similar. From there, we construct matches on individuals with similar sets of important subgraphs; here, the set of important subgraphs is learned from a training set. We generalize the Almost-Matching-Exactly (AME) framework (Dieng et al., 2019; Wang et al., 2019) to match units on subgraphs in experimental settings. We do this by constructing graph-based features that can explain both the interference pattern in the experiment and predict the underlying social network. We demonstrate that our method performs better than other methods for the same problem in many settings, while generating interpretable matches.

The paper will proceed as follows: In Section 2, we make explicit the assumptions underpinning our framework, and outline our matching approach to estimating direct treatment effects. In Sections 3 and 4, we evaluate the effectiveness of our method on simulated and real-world data. Theoretical evaluation of our approach is available in the appendix.

1.1 Related Work

Work on estimating causal effects under interference between units has three broad themes. First, there has been a growing body of work on the design of novel randomization schemes to perform causal inference under interference (Liu and Hudgens, 2014; Sinclair et al., 2012; Duflo and Saez, 2003; Basse and Airoldi, 2018). Some of this work makes explicit use of observed network structure to randomly assign treatment so as to reduce interference (Ugander et al., 2013; Toulis and Kao, 2013; Eckles et al., 2016, 2017; Jagadeesan et al., 2019). These methodologies are inapplicable to our setting as they require non-uniform treatment assignment, whereas in our setting we wish to correct for interference after randomization. Second, there is work on estimating direct treatment effects in experiments under interference, and after conventional treatment randomization, similar to our setting. Some existing work aims to characterize the behavior of existing estimators under interference (Manski, 2013; Sävje et al., 2017). Other approaches lay out methods based on randomization inference to test a variety of hypotheses under interference and treatment randomization (Rosenbaum, 2007; Aronow, 2012; Athey et al., 2018). Some of these approaches mix randomization inference and outcome models (Bowers et al., 2013). For the explicit problem of recovery of treatment effects under interference, Aronow et al. (2017) provide a general framework to translate different assumptions about interference into inverse-probability estimators, and Sussman and Airoldi (2017) give linearly unbiased, minimum integrated-variance estimators under a series of assumptions about interference. These methods either ignore explicit network structure, or require probabilities under multiple complex sampling designs to be estimated explicitly. Finally, there have been studies of observational inference under network interference (van der Laan, 2014; Liu et al., 2016; Ogburn et al., 2017; Forastiere et al., 2016). However, recovering causal estimates using observational data when units are expected to influence each other requires a structural model of both the nature of interference and contagion among units.

2 METHODOLOGY

We discuss our problem and approach in this section.

2.1 Problem Statement

We have a set of n experimental units indexed by i. These units are connected in a known graph $G = (V, E)$, where $V(G) = \{1, \dots, n\}$ is the set of vertices of G, and $E(G)$ is the set of edges of G. We disallow self-loops in our graph. We say that H is a subgraph of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Let $t_i \in \{0, 1\}$ represent the treatment indicator for unit i, $\mathbf{t}$ represent the vector of treatment indicators for the entire sample, and $\mathbf{t}_{-i}$ represent the treatment indicators for all units except i. Given a treatment vector $\mathbf{t}$ on the entire sample (i.e., for all vertices in G), we use $G^{\mathbf{t}}$ to denote the *labeled graph*, where each vertex $i \in V(G)$ has been labeled with its treatment indicator t_i. In addition, we use G_P to denote a graph induced by the set of vertices $P \subseteq V(G)$ on G, such that $V(G_P) = P$ and $E(G_P) = \{(e_1, e_2) \in E(G) :\ e_1 \in P, e_2 \in P\}$. We use the notation $\mathcal{N}_i = \{j : (i, j) \in E(G)\}$ to represent the neighborhood of vertex i. The labeled neighborhood graph of a unit i, $G^{\mathbf{t}}_{\mathcal{N}_i}$, is defined as the graph induced by the neighbors of i, and labeled according to $\mathbf{t}$. We also define $\mathbf{t}_{\mathcal{N}_i}$ to be the vector of treatment indicators corresponding to unit i's neighborhood graph. A unit's response to the treatment is represented by its random potential outcomes $Y_i(\mathbf{t}) = Y_i(t_i, \mathbf{t}_{-i})$. Unlike other commonly studied causal inference settings, unit i's potential outcomes are now a function of both the treatment assigned to i, and of all other units' treatments. Observed treatments for unit i and the whole sample are represented by the random variables T_i and $\mathbf{T}$ respectively. We assume that the number of treated units is always $n^{(1)}$, i.e., $\sum_{i=1}^{n} T_i = n^{(1)}$.

A0: Ignorability of Treatment Assignment. We make the canonical assumption that treatments are administered independently of potential outcomes, that is: $Y_i(t_i, \mathbf{t}_{-i}) \perp\!\!\!\perp \mathbf{T}$, and $0 < \Pr(T_i = 1) < 1$ for all units. In practice, we assume that treatment is assigned uniformly at random to units, which is possible only in experimental settings. As stated before, we **do not** make the canonical Stable Unit Treatment Value Assumption (SUTVA) (Rubin, 1980), which, among other requirements, states that units are exclusively affected by the treatment assigned to them. We do not make this assumption because our units are connected in a network: it could be possible for treatments to spread along the edges of the network and to affect connected units' outcomes. We do maintain the assumption of comparable treatments across units, which is commonly included in SUTVA.

Our causal quantity of interest will be the Average Direct Effect (ADE), which is defined as follows:

$$ADE = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[Y_i(1,\mathbf{0}) - Y_i(0,\mathbf{0})], \qquad (1)$$

where $\mathbf{t}_{-i} = \mathbf{0}$ represents the treatment assignment in which no unit other than i is treated. The summand represents the treatment effect on unit i when no other unit is treated, and, therefore, no interference occurs (Halloran and Struchiner, 1995).

2.2 Framework

We outline the requirements of our framework for direct effect estimation under interference. We denote interference effects on a unit i with the function $f_i(\mathbf{t}) : \{0,1\}^n \mapsto \mathbb{R}$, a function that maps each possible treatment allocation for the n units to the amount of interference on unit i. We will use several assumptions to restrict the domain of f to a much smaller set (and overload the notation f_i accordingly). To characterize f, we rely on the typology of interference assumptions introduced by Sussman and Airoldi (2017). The first three assumptions (A0-A2) needed in our framework are common in the interference literature (e.g., Manski, 2013; Toulis and Kao, 2013; Eckles et al., 2016; Athey et al., 2018):

A1: Additivity of Main Effects. First, we assume that main treatment effects are additive, i.e., that there is no interaction between units' treatment indicators. This allows us to write:

$$Y_i(t, \mathbf{t}_{-i}) = t\tau_i + f_i(\mathbf{t}_{-i}) + \epsilon_i \qquad (2)$$

where τ_i is the direct treatment effect on unit i, and ϵ_i is some baseline effect.

A2: Neighborhood Interference. We focus on a specific form of the interference function f_i by assuming that the interference experienced by unit i depends only on treatment of its neighbors. That is, if for two treatment allocations $\mathbf{t}, \mathbf{t}'$ we have $\mathbf{t}_{\mathcal{N}_i} = \mathbf{t}'_{\mathcal{N}_i}$ then $f_i(\mathbf{t}) = f_i(\mathbf{t}')$. To make explicit this dependence on the neighborhood subgraph, we will write $f_i(\mathbf{t}_{\mathcal{N}_i}) \equiv f_i(\mathbf{t})$.

A3: Isomorphic Graph Interference We assume that, if two units i and j have *isomorphic labeled neighborhood graphs*, then they receive the same amount of interference, denoting isomorphism by $\simeq$, $G^{\mathbf{t}}_{\mathcal{N}_i} \simeq G^{\mathbf{t}}_{\mathcal{N}_j} \implies f_i(\mathbf{t}_{\mathcal{N}_i}) = f_j(\mathbf{t}_{\mathcal{N}_j}) \equiv f(G^{\mathbf{t}}_{\mathcal{N}_i}) = f(G^{\mathbf{t}}_{\mathcal{N}_j})$. While Assumptions A1 and A2 are standard, A3 is new. This assumption allows us to study interference in a setting where units with similar neighborhood subgraphs experience similar amounts of interference.

All our assumptions together induce a specific form for the potential outcomes, namely that they depend on neighborhood structure $G^{\mathbf{t}}_{\mathcal{N}_i}$, but not exactly who the neighbors are (information contained in $\mathcal{N}_i$) nor treatment assignments for those outside the neighborhood (information contained in $\mathbf{t}_{\mathcal{N}_i}$). Namely:

Proposition 1. *Under assumptions A0-A3, potential outcomes in (2) for all units i can be written as:*

$$Y_i(t, \mathbf{t}_{-i}) = t\tau_i + f(G^{\mathbf{t}}_{\mathcal{N}_i}) + \epsilon_i, \qquad (3)$$

where τ_i is the direct treatment effect on unit i, and ϵ_i is some baseline response.

In addition, suppose that baseline responses for all units are equal to each other in expectation, i.e., for all i, $\mathbb{E}[\epsilon_i] = \alpha$. Then under assumptions A0-A3, for neighborhood graph structures g_i of unit i and treatment vectors $\mathbf{t}$, *the ADE is identified as:*

$$ADE = \frac{1}{n^{(1)}}\sum_{i=1}^{n}\mathbb{E}\big[T_i \times \big(\mathbb{E}[Y_i|G^{\mathbf{T}}_{\mathcal{N}_i} \simeq g_i^{\mathbf{t}}, T_i = 1] - \mathbb{E}[Y_i|G^{\mathbf{T}}_{\mathcal{N}} \simeq g_i^{\mathbf{t}}, T_i = 0]\big)\big],$$

where $G^{\mathbf{T}}_{\mathcal{N}_i}$ is the neighborhood graph of i labelled according to the treatment assignment $\mathbf{T}$.

The proposition (whose proof is in the appendix) states that the interference received by a unit is a function of each unit's neighborhood graph. Further, the outcomes can be decomposed additively into this function and the direct treatment effect on i. The proposition implies that the ADE is identified by matching each treated unit to one or more control units with an isomorphic neighborhood graph, and computing the direct effect on the treated using these matches. This effect is, in expectation over individual treatment assignments, equal to the ADE.

2.3 Subgraph Selection via Almost-Matching-Exactly

Given Proposition 1 and the framework established in the previous section, we would ideally like to match treated and control units that have isomorphic neighborhood graphs. This would allow us to better estimate the ADE without suffering interference bias: for a treated unit i, if a control unit j can be found such that $G^{\mathbf{t}}_{\mathcal{N}_i} \simeq G^{\mathbf{t}}_{\mathcal{N}_j}$, then j's outcome will be identical in expectation to i's counterfactual outcome and can be used as a proxy. Unfortunately, the number of non-isomorphic (canonically unique) graphs with a given number of nodes and edges grows incredibly quickly (Harary, 1994) and finding such matches is infeasible for large graphs. We therefore resort to counting all subgraphs that appear in a unit's neighborhood graph and matching units based on the counts of those subgraphs. However, instead of *exactly* matching on the counts of those subgraphs, we match treated and control units if they have *similar* counts, since matching exactly on all subgraph counts implies isomorphic neighborhoods and is also infeasible. Further, absolutely exact matches may not exist in real networks.

Constructing inexact matches, in turn, requires a measure of relative graph importance. In Figure 1, for example, there are two control units that the treated unit may be matched to; if triangles contribute more to the interference function, it should be matched to the right; otherwise, if degree and/or two-stars are more important, it should be matched to the left. Of course, these relative importance measures might depend on the problem and we would like to learn them.

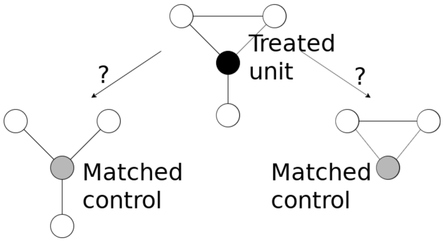

Figure 1: Inexact matching presupposes an ordering of feature importance; should the the treated ego (black) be matched to a control whose neighborhood graph has the same number of units (left), or same number of triangles (right)?

It might be tempting to match directly on f, as that would lead to unbiased inference. However, we abstain from doing so for two reasons. Firstly, in practice, the true interference is unknown and we could only match on estimated values of f; this suffers from all the problems that afflict matching on estimated propensity scores without appropriate adjustments (Abadie and Imbens, 2016) or parametric approximations (Rubin and Thomas, 1996). Such corrections or approximations do not currently exist for estimated interference functions and their development is an active area of research. Secondly, interpretability is a key component of our framework that would be lost matching on f-values; these values are scalar summaries of interference that depends on entire graphs. Estimating f well would also likely require complex and uninterpretable nonparametric methods. In Section K of the appendix, we empirically compare matching units on f-values to our subgraph matching method via simulation. The loss of interpretability associated with matching on f does not yield substantial gains in performance, even when using *true* values of f for matching, which is impossible in practice.

Almost-Matching-Exactly (AME) (Wang et al., 2019; Dieng et al., 2019; Awan et al., 2019) provides a framework for the above problem that is explicitly geared towards building interpretable, high-quality matches on discrete covariates, which in our setting are the counts of the treated subgraphs in the neighborhood. AME performs inexact matching while *learning* importance weights for each covariate from a training set, prioritizing matches on more important covariates. In this way, it neatly addresses the challenge of inexact matching by learning a metric specific to discrete covariates (namely, a weighted Hamming distance). Formally, AME matches units so as to optimize a flexible measure of match quality. For each treated unit i, solving the AME problem is equivalent to finding:

$$\boldsymbol{\theta}^{i^*} \in \underset{\boldsymbol{\theta}\in\{0,1\}^p}{\arg\max}\, \boldsymbol{\theta}^T \mathbf{w} \quad (4)$$
$$\text{such that } \exists j : t_j = 0 \text{ and } \mathbf{x}_j \circ \boldsymbol{\theta} = \mathbf{x}_i \circ \boldsymbol{\theta}$$

where $\circ$ denotes the Hadamard product, $\mathbf{w}$ is a vector of weights and $\boldsymbol{x}_i, \boldsymbol{x}_j$ are vectors of binary covariates for units i and j that we might like to match on. In our network interference setting, these are vectors of subgraph counts. The vector $\mathbf{w}$ denotes the importance of each subgraph in causing interference. We will leverage both information on outcomes and networks to construct an estimate for it.

We start by enumerating (up to isomorphism) all the p subgraphs $g_1, \dots, g_p$ that appear in any of the $G^{\mathbf{t}}_{\mathcal{N}_i}, i \in 1, \dots, n$. The covariates for unit i are then given by $S(G^{\mathbf{t}}_{\mathcal{N}_i}) = (S_1(G^{\mathbf{t}}_{\mathcal{N}_i}), \dots, S_p(G^{\mathbf{t}}_{\mathcal{N}_i}))$ where $S_k(G^{\mathbf{t}}_{\mathcal{N}_i})$ denotes the number of times subgraph g_k appears in the subgraphs of $G^{\mathbf{t}}_{\mathcal{N}_i}$. These counts are then converted into binary indicators that are one if the count of subgraph g_k in each unit's neighborhood is exactly x, for all x observed in the data. Thus, units will be matched exactly if they have identical subgraph counts. We then approximately solve the problem in Equation (4) to find the optimally important set of subgraphs upon which to exactly match each treated unit, such that there is at least one control unit that matches exactly with the treated unit on the chosen subgraph counts. The key idea behind this approach is that we want to match units exactly on subgraph counts that contribute significantly to the interference function, trading off exactly-matching on these important subgraphs with potential mismatches on subgraphs that contribute less to interference.

In practice, our implementation enumerates all subgraphs in each unit's neighborhood and stores the count of each pattern – this is computationally challenging. There is a growing body of work on efficient counting algorithms for pre-specified small patterns (up to 4-5 nodes) but there is little research on fast methods to both enumerate and count all motifs in a graph (e.g., Pinar et al., 2017; Marcus and Shavitt, 2010; Hu et al., 2013). Empirically, we see that this enumeration takes less than 30 seconds for 50 units.

The FLAME Algorithm for AME. The Fast Large Almost Matching Exactly (FLAME) algorithm (Wang et al., 2019) approximates the solution to the AME problem. The procedure starts by exactly matching all possible units on all covariates. It then drops one covariate at a time, choosing the drop maximizing the match quality MQ at that iteration, defined:

$$\mathtt{MQ} = C \cdot \mathtt{BF} - \widehat{\mathtt{PE}}_Y. \tag{5}$$

The match quality is the sum of a balancing factor BF and a predictive error $\widehat{\mathtt{PE}}_Y$, with relative weights determined by the hyper-parameter C. The balancing factor is defined as the proportion of treated units plus the proportion of control units matched at that iteration. Introducing the balancing factor into the objective has the advantage of encouraging more units to be matched, thereby minimizing variance of estimators (see Wang et al., 2019). In our setting, the second component of the match quality, predictive error, takes the form:

$$\widehat{\mathtt{PE}}_Y = \underset{h \in \mathcal{F}_1}{\arg\min} \sum_{i}^{n} (Y_i - h(S(G^{\mathbf{t}}_{\mathcal{N}_i}) \circ \boldsymbol{\theta}, T_i))^2 \tag{6}$$

for some class of functions $\mathcal{F}_1$. It is computed using a holdout training set and discourages dropping covariates that are useful for predicting the outcome. In this way, FLAME strikes a balance between matching many units and ensuring these matches are of high-quality. By using a holdout set to determine how useful a set of variables is for out-of-sample prediction, FLAME learns a measure of covariate importance via a weighted Hamming distance. Specifically, it learns a vector of importance weights $\mathbf{w}$ for the different subgraph counts that minimizes $\mathbf{w}^T \mathbb{I}[S(G^{\mathbf{t}}_{\mathcal{N}_i}) \neq S(G^{\mathbf{t}}_{\mathcal{N}_j})]$, where $\mathbb{I}[S(G^{\mathbf{t}}_{\mathcal{N}_i}) \neq S(G^{\mathbf{t}}_{\mathcal{N}_j})]$ is a vector whose k^{th} entry is 0 if the labeled neighborhood graphs of i and j have the same count of subgraph k, and 1 otherwise.

To this match quality term, we add a *network fit* term to give subgraphs more weight that are highly predictive of overall network structure. We fit a logistic regression model in which the edges (i, j) between units i, j are independent given $\mathcal{N}_i, \mathcal{N}_j$, and dependent on the subgraph counts of units i and j:

$$(i, j) \overset{iid}{\sim} \text{Bern}(\text{logit}(\beta_1^T S(G^{\mathbf{t}}_{\mathcal{N}_i}) + \beta_2^T S(G^{\mathbf{t}}_{\mathcal{N}_j})))$$

To the match quality in the original formulation, we then add $\widehat{\mathtt{PE}}_G$, defined to be the *AIC* (Akaike, 1974) of this fitted model, weighted by a hyperparameter D. Therefore, at each iteration:

$$\mathtt{MQ} = C \cdot \mathtt{BF} - \widehat{\mathtt{PE}}_Y + D \cdot \widehat{\mathtt{PE}}_G.$$

Thus, we penalize not only subgraph drops that impede predictive performance or making matches, but also those that make the observed network unlikely. $\widehat{\mathtt{PE}}_G$ represents the empirical prediction error of the chosen set of statistics for the observed graph: if $\widehat{\mathtt{PE}}_G$ is low, then the chosen subgraphs do a good job of predicting the observed graph. This error is also evaluated at a minimum over another class of prediction functions, $\mathcal{F}_2$. This term in the AME objective is justified by Assumption A3: units that have isomorphic labeled neighborhood graphs should experience the same amount of interference, and subgraph counts should be predictive of neighborhood graph structure.

Our approach to estimating the ADE is therefore as follows. (1) For each unit i, count and label all types of subgraphs in $G^{\mathbf{t}}_{\mathcal{N}_i}$. (2) Run FLAME, encouraging large numbers of matches on subgraph counts, while using the covariates that are most important for predicting the outcome and the network. (3) Estimate ADE as $\widehat{ADE}$, by computing the difference in means for each matched group and then averaging across matched groups, weighted by their size. Since our approach is based on FLAME, we call it *FLAME-Networks.*

Extensions. FLAME-Networks immediately extends to handling unit-level covariate information for baseline adjustments; we simply concatenate subgraph information and covariate information in the same dataset and then make almost-exact matches. FLAME-Networks will automatically learn the weights of both subgraphs and baseline covariates to make matches that take both into account. Another straightforward extension considers interference not just in the immediate neighborhood of each unit, but up to an arbitrary number of hops away. To extend FLAME-Networks in this way, it is sufficient to enumerate subgraphs in the induced neighborhood graph of each unit i where the vertices considered are those with a path to i that is at most k steps long. Given these counts, our method proceeds exactly as before.

d_i:	The treated degree of unit i
Δ_i:	The number of triangles in $G^{\mathbf{t}}_{\mathcal{N}_i \cup \{i\}}$ with at least one treated unit.
$\bigstar_i^k$:	The number of k-stars in $G^{\mathbf{t}}_{\mathcal{N}_i \cup \{i\}}$ with at least one treated unit.
$\dagger_i^k$:	The number of units in $G^{\mathbf{t}}_{\mathcal{N}_i \cup \{i\}}$ with degree $\geq k$ and at least one treated unit among their neighbors.
B_i:	The vertex betweenness of unit i.
C_i:	The closeness centrality of unit i.

Table 1: Interference components used in experiments; see the appendix for more details.

3 EXPERIMENTS

We begin by evaluating the performance of our estimator in a variety of simulated settings, in which we vary the form of the interference function. We find that our approach performs well in many distinct settings. In Section M of the appendix, we also assess the quality of the matches constructed.

We simulate graphs from an Erdős-Rènyi model: $G \sim$ Erdős-Rènyi(n, q), by which every possible edge between the n units is created independently, with probability q. In Sections H and I of the appendix, we also perform experiments on cluster-randomized and real-world networks. Treatments for the whole sample are generated with $\Pr(\mathbf{T} = \mathbf{t}) = \binom{n}{n^{(1)}}^{-1}$, where $n^{(1)}$ is the number of treated units. Outcomes are generated according to $Y_i(t, \mathbf{t}_{-i}) = \mathbf{t}\tau_i + f(G^{\mathbf{t}}_{\mathcal{N}_i}) + \epsilon_i$, where $\epsilon \sim N(\mathbf{0}, I_n)$ represents a baseline outcome; $\tau_i \sim N(\mathbf{5}, I_n)$ represents a direct treatment effect, and f is the interference function. In Section J of the appendix, we consider a setting in which the errors are heteroscedastic. For the interference function, we use additive combinations of the subgraph components in Table 1 and define $m_{ip}, p = 1, \ldots, 7$ to be the counts of feature p in $G^{\mathbf{t}}_{\mathcal{N}_i \cup \{i\}}$. Lastly, the counts of each component are normalized to have mean 0 and standard deviation 1. We compare our approach with different methods to estimate the ADE under interference:

Naïve. The simple difference in means between treatment and control groups assuming no interference.

All Eigenvectors. Eigenvectors for the entire adjacency matrix are computed with every treated unit matched to the control unit minimizing the Mahalanobis distance between the eigenvectors, weighing the k'th eigenvector by $1/k$. The idea behind this estimator is that the eigendecomposition of the adjacency matrix encodes important information about the network and how interference might spread within it.

First Eigenvector. Same as **All Eigenvectors** except units are matched only on their values of the largest-eigenvalue eigenvector.

Feature Weight	d_i γ_1	Δ_i γ_2	$\bigstar_i^2$ γ_3	$\bigstar_i^4$ γ_4	$\dagger^3$ γ_5	B_i γ_6	C_i γ_7
Setting 1	0	10	0	0	0	0	0
Setting 2	10	10	0	0	0	0	0
Setting 3	0	10	1	1	1	1	-1
Setting 4	5	1	10	1	1	1	-1

Table 2: Settings for Experiment 1.

Stratified Naïve. The stratified naïve estimator as discussed by Sussman and Airoldi (2017). A weighted difference-in-means estimator where units are divided into strata defined by their treated degree (number of treated vertices they are connected to), and assigned weight equal to the number of units within the stratum in the final difference of weighted averages between treated and control groups.

SANIA MIVLUE. The minimum integrated variance, linear unbiased estimator under assumptions of symmetrically received interference and additivity of main effects, when the priors on the baseline outcome and direct treatment effect have no correlation between units; proposed by Sussman and Airoldi (2017).

FLAME-Networks. Our proposed method. In all simulations, the two components of the PE function are weighted equally, and a ridge regression is used to compute outcome prediction error.

3.1 Experiment 1: Additive Interference

First we study a setting in which interference is an additive function of the components in Table 1. Outcomes in this experiment have the form: $Y_i = \gamma_1 d_i + \gamma_2 \Delta_i + \gamma_3 \bigstar_i^2 + \gamma_4 \bigstar_i^4 + \gamma_5 \dagger_i^3 + \gamma_6 B_i + \gamma_7 C_i + \epsilon_i$, with $\epsilon_i \sim N(0, 1)$. We simulate 50 datasets for each setting, in which the units are in an $ER(50, 0.05)$ graph. Table 2 shows values for the γ_i in each of our experimental settings. Results for Experiment 1 are reported in Figure 2. FLAME-Networks outperforms all other methods both in terms of average error, and standard deviation over the simulations. This is likely because FLAME-Networks learns weights for the subgraphs that are proportionate to those we use at each setting, and matches units on subgraphs with larger weights. When the interference function is multiplicative instead of additive, FLAME-Networks performs similarly; results are in the appendix.

3.2 Experiment 2: Covariate Adjustment

A strength of FLAME-Networks is its ability to natively account for covariate information. We analyze a setting in which baseline effects are dependent on an additional discrete-valued covariate, x, that is observed alongside the network. Outcomes take the form $Y_i = t\tau_i + f(G^{\mathbf{t}}_{\mathcal{N}_i}) + \beta x_i + \epsilon_i$, where x_i is chosen uni-

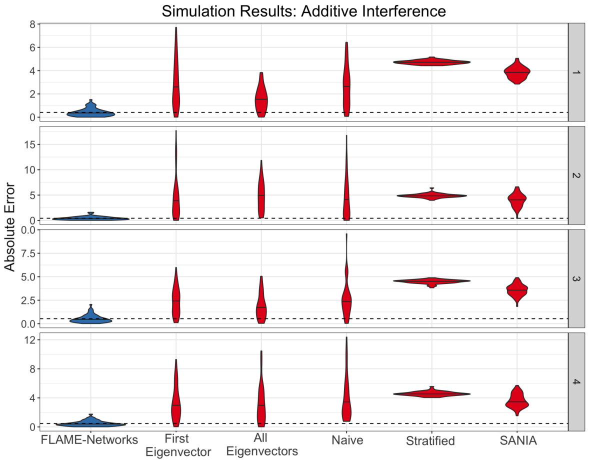

Figure 2: Results from Experiment 1. Each violin plot represents the distribution over simulations of absolute estimation error. The panels are numbered according to the parameter settings of the simulations. Violin plots are blue if the method had mean error lower than or equal to FLAME-Networks' and red otherwise. The black line inside each violin is the median error. The dashed line is FLAME-Networks' mean error.

formly at random from $\{1, 2, 3\}$ for each unit, and β is fixed at 15. This means that our sample is divided into 3 strata defined by the covariate values. We ran FLAME-Networks with a dataset consisting of subgraph counts, plus observed covariates for each unit. For comparison with the other methods, we first regress Y on x, and then use the residuals of that regression as outcomes for the other methods. This way, the initial regression will account for the baseline effects of x, and the residuals contain only potential interference. The interference function takes the form $f(G^{\mathbf{t}}_{\mathcal{N}_i}) = d_i + \Delta_i + B_i$, which is what we are trying to learn with the other methods. We simulate the sample network from $ER(50, 0.05)$.

Results are displayed in Table 3. FLAME-Networks performs, on average, better than all the other methods. Results in Section L of the supplement show that when β is increased, none of the methods suffer in performance. While regression adjustment prior to estimation seems to have a positive impact on the performance of other methods in the presence of additional covariates, FLAME-Networks performs best. This is because FLAME-Networks is built to easily handle the inclusion of covariates in its estimation procedure.

Method	Median	25th q	75th q
FLAME-Networks	0.39	0.21	0.59
First Eigenvector	0.47	0.40	0.83
All Eigenvectors	0.55	0.29	0.79
Naive	0.53	0.36	0.92
SANIA	1.93	1.75	2.25
Stratified	4.49	4.45	4.53

Table 3: Results from Experiment 2 with $\beta = 5$. Median and 25th and 75th percentile of absolute error over 40 simulated datasets.

3.3 Experiment 3: Misspecified Interference

We now study the robustness of our estimator in a setting in which one of our key assumptions – A3 – is violated. Specifically, we now allow for treated and control units to receive different amounts of interference, *even if they have the same labelled neighborhood graphs.* We do this by temporarily eliminating all control-control edges in the network and then counting the features in Table 1 used to assign interference. That is, consider a unit i with a single, untreated neighbor j. In our new setting, if degree is a feature of the interference function, then i being treated implies i receives interference from j. But if i is untreated, then i would receive no interference from j, because its neighbor is also untreated. This crucially implies that FLAME-Networks will be matching–and estimating the ADE from–individuals that do not necessarily receive similar amounts of interference.

In this setting, we generate interference according to: $f_i = (5 - \gamma)d_i + \gamma\Delta_i$ for $\gamma \in [0, 5]$ and assess the performance of FLAME-Networks against that of the SANIA and stratified estimators. Results are shown in Figure 3. We see that, when degree is the only component with weight in the interference function, FLAME-Networks performs better than the stratified estimator, but worse than the SANIA estimator, which leverages aspects of the graph related to degree. However, our performance improves as γ increases and the true interference depends more on triangle counts since the triangle counts available to FLAME-Networks represent the actual interference pattern more frequently than the degree counts did. Thus, we see that although violation of our method's assumptions harms its performance, it still manages at times to outperform estimators that rely too heavily on degree.

4 APPLICATION

In this section, we demonstrate the practical utility of our method. We use data collected by Banerjee et al. (2013) on social networks for 75 villages in Karnataka, India. They are a median distance of 46 km from one another other, motivating the assumption that network interference is experienced solely between individuals from the same village. For each village, we study the effect of 1. lack of education on election participation; 2. lack of education on Self-Help-Group

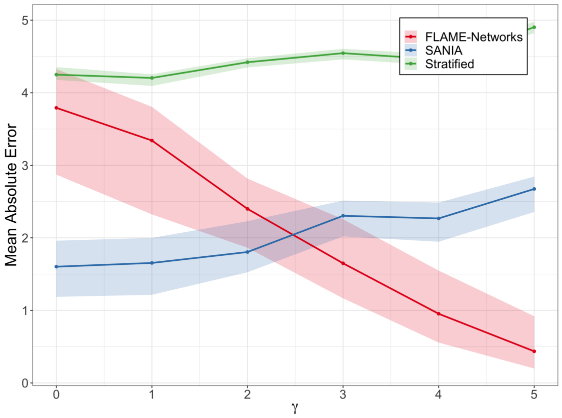

Figure 3: Results from Experiment 3. These simulations were run on an ER(75, 0.07) graph. The bands around the lines represent 25th and 75th quantiles of the 50 simulations, for each value of γ.

(SHG) participation; and 3. being male on SHG participation. We proxy election participation by ownership of an election card. We compare our estimates – which account for network interference – to naive estimates – which assume no network interference. Data pre-processing is summarized in the appendix.

For ADE estimates, we assume the treatment is randomly assigned. We find that lack of education is associated with higher SHG participation, and that males are less likely to participate in SHGs than females (see Figure 4). These results make sense in the context of developing countries where SHGs are mainly utilized by females in low-income families, which generally have lower education levels. We observe that education does not impact election participation; in developing countries, an individual's decision to participate in an election may be driven by factors such caste, religion, influence of local leaders and closeness of race (Shachar and Nalebuff, 1999; Gleason, 2001). FLAME-Networks matches units in each village by subgraph counts and covariate values to estimate the ADE. Looking at the matched groups, we discover that subgraphs such as 2-stars and triangles were important for matching, implying that second-order connections could be affecting interference in this setting. Further details of the matched groups are in Section F.

Figure 4 plots naive and FLAME-Networks ADE estimates. We find a significant difference between our estimates and the naive estimates when estimating the effect of being male on participation in SHGs. The naive estimator overestimates the treatment effect, which is possible when ignoring network interference. It is plausible, in this setting, that interference would heighten these effects for all outcomes. This is because individuals from similar social backgrounds or gender tend to interact more together, and, therefore, are more likely to influence each other's participation decision, both in elections and in SHGs.

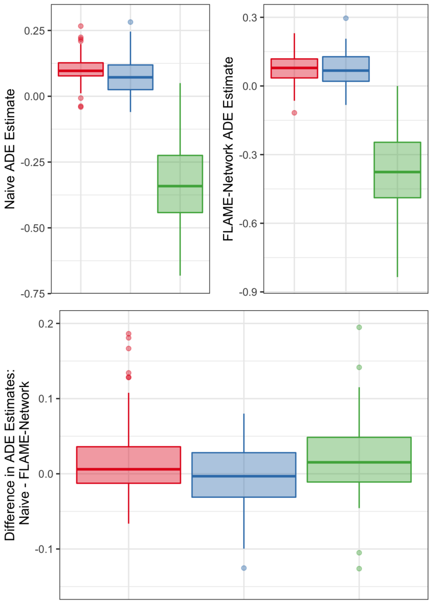

Figure 4: Naive and FLAME-Networks ADE estimates and their difference. Red, blue, and green respectively correspond to (treatment, outcome) pairs: (no education, election participation), (no education, SHG participation), and (gender, SHG participation).

5 DISCUSSION

Conventional estimators for treatment effects in randomized experiments will be biased when there is interference between units. We have introduced FLAME-Networks – a method to recover direct treatment effects in such settings. Our method is based on matching units with similar neighborhood graphs in an almost-exact way, thus producing interpretable, high-quality results. We have shown that FLAME-Networks performs better than existing methods both on simulated data, and we have used real-world data to show how it can be applied in a real setting. Our method extends easily to settings with additional covariate information for units and to taking into account larger neighborhoods for interference. In future work, our method can be extended to learning a variety of types of distance metrics between graphs.

Acknowledgements

This work was supported in part by NIH award 1R01EB025021-01, NSF awards IIS-1552538 and IIS1703431, a DARPA award under the L2M program, and a Duke University Energy Initiative ERSF grant.

References

Alberto Abadie and Guido Imbens. Matching on the estimated propensity score. *Econometrica*, pages 781–807, 2016.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974. doi: 10.1109/TAC.1974.1100705.

Peter M Aronow. A general method for detecting interference between units in randomized experiments. *Sociological Methods & Research*, 41(1):3–16, 2012.

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

Susan Athey, Dean Eckles, and Guido W Imbens. Exact p-values for network interference. *Journal of the American Statistical Association*, 113(521):230–240, 2018.

M Awan, Yameng Liu, Marco Morucci, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. Interpretable almost matching exactly with instrumental variables. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.

Abhijit Banerjee, Arun G Chandrasekhar, Esther Duflo, and Matthew O Jackson. The diffusion of microfinance. *Science*, 341(6144):1236498, 2013.

Guillaume W Basse and Edoardo M Airoldi. Model-assisted design of experiments in the presence of network-correlated outcomes. *Biometrika*, 105(4): 849–858, 2018.

Jake Bowers, Mark M Fredrickson, and Costas Panagopoulos. Reasoning about interference between units: A general framework. *Political Analysis*, 21(1):97–124, 2013.

David Roxbee Cox. *Planning of Experiments*. Wiley, 1958.

Awa Dieng, Yameng Liu, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. Interpretable almost-exact matching for causal inference. In *Proceedings of Artificial Intelligence and Statistics (AISTATS)*, pages 2445–2453, 2019.

Esther Duflo and Emmanuel Saez. The role of information and social interactions in retirement plan decisions: Evidence from a randomized experiment. *The Quarterly Journal of Economics*, 118(3):815–842, 2003.

Dean Eckles, René F Kizilcec, and Eytan Bakshy. Estimating peer effects in networks with peer encouragement designs. *Proceedings of the National Academy of Sciences*, 113(27):7316–7322, 2016.

Dean Eckles, Brian Karrer, and Johan Ugander. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1), 2017.

Laura Forastiere, Edoardo M Airoldi, and Fabrizia Mealli. Identification and estimation of treatment and interference effects in observational studies on networks. *arXiv preprint arXiv:1609.06245*, 2016.

Suzanne Gleason. Female political participation and health in india. *The ANNALS of the American Academy of Political and Social Science*, 573(1):105–126, 2001.

M Elizabeth Halloran and Claudio J Struchiner. Causal inference in infectious diseases. *Epidemiology (Cambridge, Mass.)*, 6(2):142–151, 1995.

Frank Harary. *Graph Theory*. Addison-Wesley, 1994.

Kathleen Harris. The add health study: design and accomplishments. Technical report, Carolina Population Center: University of North Carolina at Chapel Hill, 2013.

Xiaocheng Hu, Yufei Tao, and Chin-Wan Chung. Massive graph triangulation. In *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*, pages 325–336, 2013.

Ravi Jagadeesan, Natesh Pillai, and Alexander Volfovsky. Designs for estimating the treatment effect in networks with interference. *Annals of Statistics*, 2019.

Lan Liu and Michael G Hudgens. Large sample randomization inference of causal effects in the presence of interference. *Journal of the American Statistical Association*, 109(505):288–301, 2014.

Lan Liu, Michael G Hudgens, and Sylvia Becker-Dreps. On inverse probability-weighted estimators in the presence of interference. *Biometrika*, 103(4): 829–842, 2016.

Charles F Manski. Identification of treatment response with social interactions. *The Econometrics Journal*, 16(1):S1–S23, 2013.

Dror Marcus and Yuval Shavitt. Efficient counting of network motifs. In *2010 IEEE 30th International Conference on Distributed Computing Systems Workshops*, pages 92–98. IEEE, 2010.

Elizabeth L Ogburn, Tyler J VanderWeele, et al. Vaccines, contagion, and social networks. *The Annals of Applied Statistics*, 11(2):919–948, 2017.

Ali Pinar, C Seshadhri, and Vaidyanathan Vishal. Escape: Efficiently counting all 5-vertex subgraphs. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1431–1440, 2017.

Paul R Rosenbaum. Interference between units in randomized experiments. *Journal of the American Statistical Association*, 102(477):191–200, 2007.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Donald Rubin and Neal Thomas. Matching using estimated propensity scores: relating theory to practice. *Biometrics*, 52(1):249–264, 1996.

Donald B Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American Statistical Association*, 75(371):591–593, 1980.

Fredrik Sävje, Peter M Aronow, and Michael G Hudgens. Average treatment effects in the presence of unknown interference. *arXiv preprint arXiv:1711.06399*, 2017.

Ron Shachar and Barry Nalebuff. Follow the leader: Theory and evidence on political participation. *American Economic Review*, 89(3):525–547, 1999.

Betsy Sinclair, Margaret McConnell, and Donald P Green. Detecting spillover effects: Design and analysis of multilevel experiments. *American Journal of Political Science*, 56(4):1055–1069, 2012.

Michael E Sobel. What do randomized studies of housing mobility demonstrate? causal inference in the face of interference. *Journal of the American Statistical Association*, 101(476):1398–1407, 2006.

Daniel L Sussman and Edoardo M Airoldi. Elements of estimation theory for causal effects in the presence of network interference. *arXiv preprint arXiv:1702.03578*, 2017.

Panos Toulis and Edward Kao. Estimation of causal peer influence effects. In *International Conference on Machine Learning (ICML)*, pages 1489–1497, 2013.

Johan Ugander, Brian Karrer, Lars Backstrom, and Jon Kleinberg. Graph cluster randomization: Network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 329–337. ACM, 2013.

Mark J van der Laan. Causal inference for a population of causally connected units. *Journal of Causal Inference*, 2(1):13–74, 2014.

Tianyu Wang, Marco Morucci, M Awan, Yameng Liu, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. FLAME: A fast large-scale almost matching exactly approach to causal inference. *arXiv preprint arXiv:1707.06315*, 2019.