Journal of Data Science 21 (3), 523–537
July 2023

DOI: 10.6339/23-JDS1112
Statistical Data Science

Network A/B Testing: Nonparametric Statistical Significance Test Based on Cluster-Level Permutation

Hongwei Shang[1,*], Xiaolin Shi[2], and Bai Jiang[3]

[1]*Walmart Global Tech, Search Dept, Sunnyvale, CA, US*
[2]*Snap Inc, Santa Monica, CA, US*
[3]*Citadel Securities, Chicago, IL, US*

Abstract

A/B testing is widely used for comparing two versions of a product and evaluating new proposed product features. It is of great importance for decision-making and has been applied as a golden standard in the IT industry. It is essentially a form of two-sample statistical hypothesis testing. Average treatment effect (ATE) and the corresponding p-value can be obtained under certain assumptions. One key assumption in traditional A/B testing is the *stable-unit-treatment-value assumption* (SUTVA): there is no interference among different units. It means that the observation on one unit is unaffected by the particular assignment of treatments to the other units. Nonetheless, interference is very common in social network settings where people communicate and spread information to their neighbors. Therefore, the SUTVA assumption is violated. Analysis ignoring this network effect will lead to biased estimation of ATE. Most existing works focus mainly on the design of experiment and data analysis in order to produce estimators with good performance in regards to bias and variance. Little attention has been paid to the calculation of p-value. We work on the calculation of p-value for the ATE estimator in network A/B tests. After a brief review of existing research methods on design of experiment based on *graph cluster randomization* and different ATE estimation methods, we propose a permutation method for calculating p-value based on permutation test at the cluster level. The effectiveness of the method against that based on individual-level permutation is validated in a simulation study mimicking realistic settings.

Keywords *design of experiments; graph cluster randomization; p-value*

1 Introduction

A/B testing is conducted when two variants of a product, A and B, need to be compared against each other (e.g., Kohavi et al., 2013). It is widely used for evaluating new proposed product features. It is important for decision-making and has been applied as a golden standard in the IT industry. Before releasing a new version of features to the entire users of a product, a sample are randomly selected for an online experiment for A/B testing. The *experimental units* are randomly assigned to two variants of the test — version A (the *control* group) and version B (the *treatment* group). Then each experimental unit is exposed to the assigned variants of the experiment for some period of time. User identifier, such as cookie, is commonly used as an

*Corresponding author. Email: hongwei.shang@walmart.com.

Received July 14, 2023; Accepted July 14, 2023

experimental unit on the web, and in this paper we will use user as our experimental unit for illustration purpose.

In order to see which version of the variants is better, an *overall evaluation criterion* (OEC) is defined as a quantitative measure of the experiment's objective, which is also called the *response variable* in statistics (Kohavi et al., 2012). The difference between the response values for two groups is the *treatment effect*, sometimes also called total treatment effect. To estimate the treatment effect, the response values are averaged across all users for each group; the difference between the averaged response values for two groups is the estimated *treatment effect*. This treatment effect is *statistically significant* if the test rejects the null hypothesis that the response values are not different. The null hypothesis is rejected at a pre-chosen significance level α when the p-value $p < \alpha$ and not rejected when $p > \alpha$. As a convenient standard statistical measurement, p-value has been widely used in industry for helping decision makers to evaluate online bucket performance and make reliable product decisions.

One key assumption in traditional A/B testing comparing two treatments is that there is no interference between different units, which means that the observation on one unit is unaffected by the particular assignment of treatments to the other units. This is called the *stable-unit-treatment-value assumption*, or SUTVA (Rubin, 1986). For example, assuming we are conducting an A/B testing for font size of search queries in a search engine, a sample of users are randomly assigned to control group (who see default font size) and treatment group (who see new font size). Since there is no interference among users, a user i's response when assigned to control group under current experimental design is the same as what would be observed under global control; likewise, a user's response when assigned to treatment group is the same as what would be observed under global treatment.

Interference is unfortunately a common occurrence in social network settings where people communicate and share information with their connections. In the context of A/B testing on social networks, the SUTVA assumption is violated. Let us consider a scenario where we are testing a feature in a social network, and a user's response is heavily influenced by the behavior of their neighbors. If a user i is assigned to the treatment group, there will be a significant difference in user i's expected outcome depending on whether their neighbors are assigned to the control group or if they are all included in the treatment group. In other words, the response of user i is not only affected by their own assignment but also by their neighbors' assignments.

Next let us illustrate why SUTVA is an important assumption in A/B testing. For N experiment units ($i = 1, \ldots, N$), let $\boldsymbol{z} = \{z_1, \ldots, z_N\}$ be the treatment assignment vector with $z_i \in \{0, 1\}$ ($i = 1, \ldots, N$), where $z_i = 0$ means the user i is in the control group and $z_i = 1$ means the user i is in the treatment group. Let $\boldsymbol{Y}(\boldsymbol{z}) = \{Y_1(\boldsymbol{z}), \ldots, Y_N(\boldsymbol{z})\}$ be the potential response values vector under the treatment assignment vector $\boldsymbol{z}$, where $Y_i(\boldsymbol{z})$ be the potential response value for user i under the treatment assignment vector $\boldsymbol{z}$. The key quantity in A/B testing is the average treatment effect (ATE) of applying all users in global treatment ($\boldsymbol{z} = \vec{1}$) compared to applying all users in global control ($\boldsymbol{z} = \vec{0}$):

$$\tau(\boldsymbol{z} = \vec{1}, \boldsymbol{z} = \vec{0}) = \frac{1}{N} \sum_{i=1}^{N} \big[Y_i(\boldsymbol{z} = \vec{1}) - Y_i(\boldsymbol{z} = \vec{0}) \big]. \tag{1}$$

In reality, we are not able to observe each unit under both treatment and control case. Under SUTVA, the value of Y for unit i when exposed to treatment z_i will be the same no matter what treatments other units receive, which holds for all experiment units and both groups. Thus, for each user i, the possible response values can be represented by Y_{i,z_i} ($i = 1, \ldots, N$; $j = 0, 1$),

depending only on the user's own treatment assignment. Thus, in the traditional A/B testing framework, the ATE in (1) can be simplified to

$$\tau(z = \vec{1}, z = \vec{0}) = \left(\sum_{i=1}^{N} Y_{i,1} - \sum_{i=1}^{N} Y_{i,0}\right)/N \tag{2}$$

The common solution for classic A/B testing is to randomly assign users to two groups. Assume N_1 and N_0 users are assigned to treatment and control group respectively, then we can use

$$\hat{\tau} = \frac{1}{N_1} \sum_{\{i:z_i=1\}} Y_{i,z_i} - \frac{1}{N_0} \sum_{\{i:z_i=0\}} Y_{i,z_i}$$

to estimate the ATE. This is a reasonable approximation under SUTVA assumption. However, in social network setting, SUTVA is not valid due to interference among users. The same analysis ignoring this interference will bring bias for ATE estimators.

There has been a lot of research on A/B testing in recent years based on practical lessons learned from industry experience (see, e.g., Kohavi et al., 2020, 2014, 2012, 2010). In contrast to traditional A/B testing analysis which is pretty mature, network A/B testing gained sharpened focus only relatively recently (see, e.g., Gui et al., 2015; Saveski et al., 2017). Most research work focuses on restricting assumptions of network exposure, sampling techniques, experimental design and estimator performance. Various sampling techniques have been developed to produce internally well-connected but also approximately uniformly distributed nodes over the population (see, e.g., Backstrom and Kleinberg, 2011; Katzir et al., 2012). Different clustering methodologies were proposed for experimental design based on *graph cluster randomization*, and different estimators were proposed and evaluated compared based on bias and variance (Eckles et al., 2014; Karrer et al., 2021; Liu et al., 2022; Ugander and Yin, 2023). However, there is little work on the statistical significance test. If a closed-form distribution for the ATE estimator under the null hypothesis exists, p-value for the tests can be obtained theoretically. For example, in classical A/B testing, for a two sample t-test or binomial test, the sampling distribution is approximately normal when sample size is large enough. For network A/B testing, such parametric forms do not exist due to interference among users.

A promising alternative approach to network A/B testing is to use nonparametric permutation methods. The advantages of nonparametric tests based on permutation for network A/B testing were first proposed by Jiang et al. (2016) who applied the permutation test for test statistics computed from the Ising model. Yet this permutation test at the individual level has caveats under experimental design based on *graph cluster randomization*; we will go through this definition in the paper. We circumvented this issue by proposing a permutation test at the cluster level. Starting from the fact that users are assigned to treatment/control group at the cluster level from the design of experiment, it is natural to base permutation tests under consideration on clustering. More specifically, given clustering under experimental design, permutation conducted at cluster level ensures that users in the same cluster will always be assigned to the same variant in each permutation. To the best of our knowledge, this is the first study to propose a nonparametric permutation test considering the graph structure in network A/B testing framework. It also represents the first comparative appraisal of permutation tests at the cluster level vs at the individual level in network A/B testing via intensive simulation study for different types of test statistics.

The paper is organized as follows. Section 2 includes an overview of the phases in network A/B testing. Section 3 presents our proposed procedures for testing the null hypothesis of no

treatment effect. Section 4 reports the results of simulation study with data generated based on small world graph under different rewiring probabilities. The last section concludes.

2 Network A/B Testing

2.1 Network Exposure

In social network setting where interference exists, a user's potential response value is determined not only by his own treatment assignment z_i, but also other users' treatment assignment $\boldsymbol{z}$. At the worst scenario, each specification of $\boldsymbol{z}$ will produce a unique potential response value Y_i for user i, bringing 2^N possible potential values. This so-called "arbitrary exposure" (Aronow and Samii, 2012) makes it impossible for estimating ATE, since none of users can be assumed to produce approximately the same potential response value as if the entire sample is in the same group as him/her. Thus further assumptions need to be made in order to describe users whose response under the particular treatment assignment vector is approximately the same as what would be observed under global treatment of interest.

First, the notion of *network exposure* was introduced by Ugander et al. (2013). A user i in the treatment group is defined as *network exposed* to the treatment under a particular assignment $\boldsymbol{z}$ if user i's response under $\boldsymbol{z}$ is the same as user i's response under $\boldsymbol{z} = \vec{1}$; a user i in the control group is defined as *network exposed* to the control under a particular assignment $\boldsymbol{z}$ if user i's response under $\boldsymbol{z}$ is the same as user i's response under $\boldsymbol{z} = \vec{0}$. Different scenarios could be investigated for network exposure. One basic scenario is that a user i is network exposed to the treatment if user i and all his/her neighbors are in the treatment group; user i is network exposed to control group if user i and all his/her neighbors are in the control group. Another scenario is to fix a threshold $q \in \{0, 1\}$ and define that user i is network exposed to the treatment if user i and at least $100q\%$ of i's neighbors are in the treatment group; user i is network exposed to the control if user i and at least $100q\%$ of i's neighbors are in the control group. Given a treatment assignment vector $\boldsymbol{z}$, each scenario of exposure conditions specifies users who are network exposed to treatment as if the entire sample is in treatment and users who are network exposed to control as if the entire sample is in control group.

Next, the restriction of "arbitrary exposure" is relaxed by letting multiple treatment assignment vectors $\boldsymbol{z}$ produce the same Y_i for a user i. More specifically, for user i with $z_i = 1$, we are interested in those assignment vectors producing the same Y_i as $\boldsymbol{z} = \vec{1}$; for user i with $z_i = 0$, we are interested in those assignment vectors producing the same Y_i as $\boldsymbol{z} = \vec{0}$. Thus, for user i, the particular exposure conditions Γ_i^1 and Γ_i^0 are defined by

$$\begin{aligned} \Gamma_i^1 &= \left\{\tilde{\boldsymbol{z}} | Y_i(\boldsymbol{z} = \tilde{\boldsymbol{z}}) = Y_i(\boldsymbol{z} = \vec{1})\right\} \\ \Gamma_i^0 &= \left\{\tilde{\boldsymbol{z}} | Y_i(\boldsymbol{z} = \tilde{\boldsymbol{z}}) = Y_i(\boldsymbol{z} = \vec{0})\right\}. \end{aligned} \tag{3}$$

The general exposure conditions assume that a user i's behavior depends on his/her own treatment assignment z_i and his/her adjacent users only. Further, absolute and fractional conditions on the number of neighbors in the treatment group are considered (Ugander et al., 2013). In this work we focus on the latter condition *fractional q-neighborhood exposure*, due to its robustness to the heterogeneity of users' degrees (Gui et al., 2015).

A user i is fractional q-neighborhood exposed to treatment if user i is in treatment and at least $100q\%$ of user i's neighbors are also in treatment; a user i is fractional q-neighborhood exposed to control if user i is in control and at least $100q\%$ of user i's neighbors are also in

control. Let σ_i be the percent of user i's neighbors in treatment group. Under this exposure model, given threshold q, Γ_i^1 and Γ_i^0 in (3) can be expressed specifically by

$$\begin{aligned} \Gamma_{i,q}^1 &= \{\boldsymbol{z}|z_i = 1, \sigma_i > q\} \\ \Gamma_{i,q}^0 &= \big\{\boldsymbol{z}|z_i = 0, \sigma_i < (1-q)\big\}. \end{aligned} \tag{4}$$

Given neighborhood exposure assumptions about interference, existing research work focuses on design of experiment based on *graph cluster randomization* and ATE estimation methods in order to reduce bias for the ATE.

Some network notations are needed to proceed. Let $G(V, E)$ denote graph structure, where $V = \{v_1, v_2, \ldots, v_N\}$ is a set of all vertices and E is a set of all edges in graph G, $(v_i, v_j) \in E$ if node v_i and v_j are connected. Let $\boldsymbol{A}$ denote the corresponding adjacency matrix $\boldsymbol{A}$, and vector $\boldsymbol{A}_{i,}$ is the i-th row in $\boldsymbol{A}$. Let $\boldsymbol{Z} = \{Z_1, \ldots, Z_N\}$ denote the assignment vector for all vertices; $\boldsymbol{Y} = \{Y_1, \ldots, Y_N\}$ denote response vector given $\boldsymbol{Z}$. Thus the triplet $\{G, \boldsymbol{Z}, \boldsymbol{Y}\}$ denotes a sampling process: selecting a sub-network G from the entire social network, assigning treatment/control to all vertices to obtain $\boldsymbol{Z}$, and collecting their response values $\boldsymbol{Y}$.

2.2 Graph Cluster Randomization

Researchers on network A/B testing consider clustering the graph then randomly and independently assigning treatment/control group at the cluster level instead of at the individual level, to analyze average treatment effects under network A/B testing. Then users connected to each other are more likely to be assigned to the same treatment/control group than if they are assigned independently. This generic *graph cluster randomization* scheme based on graph clustering was introduced by Ugander et al. (2013) for designing experiment of A/B testing when network effect are anticipated. Their work was motivated by producing an estimator of ATE with asymptotically small variance. Ugander et al. (2013) used r-net clustering method for the shortest-path metric of graph (Gupta et al., 2003). To build a r-net clustering, vertices v_1, v_2, ... are identified such that any two identified vertices are at least r steps from each other. Then for each w belonging to the remaining unidentified vertices, assign w to closest v in $v_1, v_2, \ldots$. Then clusters C_1, C_2, ... are formed by v_1, v_2, ... respectively. In this clustering method, since the size of each cluster depends heavily on the degree of center nodes, the cluster sizes vary with the extent depending on how the degrees of central nodes vary.

Further works investigated the properties of the graph cluster randomization. Gui et al. (2015) pointed out that this will introduce bias in the ATE estimator, and further proposed balanced graph partitioning clustering method. Saveski et al. (2017) also pointed out two main practical reasons for partitioning the graph into clusters of equal size: *variance reduction* and *balance on pre-treatment covariates*. Saveski et al. (2017) evaluated multiple balanced clustering algorithms, including METIS (Karypis and Kumar, 1998), Balanced Label Propagation (BLP) (Ugander and Backstrom, 2013), Restreaming Linear Deterministic Greedy (reLDG), and Restreaming FUNNEL (reFUNNEL) (Nishimura and Ugander, 2013). These balanced clustering algorithms were applied on the subgraph of the full LinkedIn graph, and reLDG performs the best among these balanced clustering algorithms (Saveski et al., 2017). In this paper, we also use reLDG for graph partitioning at our experiments.

Clustering randomized sampling tends to assign a user and his/her neighbors in the same treatment/control group, and is able to control for "contamination" across clusters. Under fractional q-neighbor exposure condition, graph cluster randomization puts users closer to the condition of global treatment of interest. Thus graph cluster randomization can reduce bias in

ATE estimators dramatically compared against randomization performed at individual user level without increasing variance very much; the benefit is even larger when there is strong social interactions and more local clustering in the network (Eckles et al., 2014).

2.3 Estimation

For estimating ATE, a naive estimator is a simple difference in the sample means between users in treatment group and in control group

$$\hat{\tau}_{naive} = \frac{1}{|\{i|Z_i = 1\}|} \sum_{i:Z_i=1} Y_i - \frac{1}{|\{i|Z_i = 0\}|} \sum_{i:Z_i=0} Y_i,$$

where $|\cdot|$ denotes cardinality. Estimator $\hat{\tau}_{naive}$ is unbiased under SUTVA assumption if no network effect exists. In social network settings, bias arises when users' responses are affected by his/her neighbors. The magnitude of the bias depends on how strong the network effect is.

The bias can be reduced by comparing only users who behave similarly as if they were placed in global treatment of interest. Under the fractional q-neighbor exposure model, a user i behaves as if he/she is in global treatment/control group given that user i is fractional q-neighborhood exposed to treatment/control. For users who are not q-neighborhood exposed to treatment/control, their responses are ineffective and hence removed from estimation. Thus, given the assignment vector $\boldsymbol{Z}$, ATE can be estimated by

$$\hat{\tau}_{neighbor} = \frac{1}{N_q^1} \sum_{i:\mathbf{Z}\in\Gamma_{i,q}^1} Y_i - \frac{1}{N_q^0} \sum_{i:\mathbf{Z}\in\Gamma_{i,q}^0} Y_i,$$

where N_q^1 and N_q^0 denote the number of users that q-neighborhood exposed to control, treatment respectively. From Gui et al. (2015)'s observation, the choice of q denotes a tradeoff between bias and variance. A large q close to 1 means that only users with most of neighbors in the same group as his/hers can be regarded as effective users, which leads to a small bias. Yet, the variance will be large due to the fact that few number of effective users can be used for estimation. On the contrary, a small q will lead to smaller variance but larger bias.

Estimator $\hat{\tau}_{neighbor}$ will only be unbiased for ATE if we can assume that each effective user has the same probability of being assigned to a chosen *effective treatment* (Eckles et al., 2014). In this work, a user i is in *effective treatment* if he/she is fractional q-neighborhood exposed to treatment/control. Ugander et al. (2013) observed that users with high degrees are less likely to be in effective treatment, while users with low degree are more likely to be in such effective treatment. For example, assuming $q = 0.9$, a user i with $z_i = 1$ is much more likely to be exposed to treatment if he/she has only one friend than if he/she has 100 friends. Given a clustering of all users and the pre-specified threshold q, the probability of being network exposed to treatment and to control for user i is $\Pr(\boldsymbol{Z} \in \Gamma_{i,q}^1)$ and $\Pr(\boldsymbol{Z} \in \Gamma_{i,q}^0)$. These two quantities can be computed explicitly using a dynamic program (Ugander et al., 2013). Note that the procedure of clustering users tend to put connected users in the same cluster, hence a user has higher chance to be network exposed to a condition under randomization at cluster level.

Considering this exposure probability, the allocation bias can be corrected using the Hajek estimator (Aronow and Samii, 2012) with the following form:

$$\hat{\tau}_{Hajek} = \frac{\sum_{i:\mathbf{Z}\in\Gamma_{i,q}^1} \frac{Y_i}{\Pr(\mathbf{Z}\in\Gamma_{i,q}^1)}}{\sum_{i:\mathbf{Z}\in\Gamma_{i,q}^1} \frac{1}{\Pr(\mathbf{Z}\in\Gamma_{i,q}^1)}} - \frac{\sum_{i:\mathbf{Z}\in\Gamma_{i,q}^0} \frac{Y_i}{\Pr(\mathbf{Z}\in\Gamma_{i,q}^0)}}{\sum_{i:\mathbf{Z}\in\Gamma_{i,q}^0} \frac{1}{\Pr(\mathbf{Z}\in\Gamma_{i,q}^0)}}.$$

Both $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{Hajek}$ utilizes information from a subset of users depending the choice of threshold q. Gui et al. (2015) proposed new estimators based on the fraction neighborhood exposure model. In this model, a user i's expected response function $g(Z_i, \sigma_i)$ depends on Z_i and σ_i; ATE τ can then be denoted by

$$\tau = g(1,1) - g(0,0) \tag{5}$$

correspondingly. Here all observations are utilized regardless of whether they are network exposed. Specifically, Gui et al. (2015) suggested using a linear model to estimate ATE:

$$g(Z_i, \sigma_i) = \beta_0 + \beta_1 Z_i + \beta_2 \sigma_i + \beta_3 Z_i \sigma_i, \tag{6}$$

where the regression coefficients β_0, β_1, β_2, and β_3 are estimated as $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{\beta}_3$ by regressing $g(Z_i, \sigma_i)$ on Z_i and σ_i. Their first model fixes β_3 at $\beta_3 = 0$, assuming no interaction between Z_i and σ_i, and the ATE can be estimated by $\hat{\tau}_{lm_1} = \hat{\beta}_1 + \hat{\beta}_2$. Their second model includes interaction between Z_i and σ_i, and the estimated ATE is $\hat{\tau}_{lm_2} = \hat{\beta}_1 + \hat{\beta}_2 + \hat{\beta}_3$.

3 Significance Tests Based on Cluster-Level Permutation

In Section 2, the process of experiments design based on graph cluster randomization and estimation methods were described, both of which can reduce bias and error for estimating ATE. Further, in order to make reliable product decisions for decision-makers, the two-sample statistical hypothesis testing needs to be conducted. The specification of the null hypothesis H_0 is: H_0: ATE $= 0$, vs. H_a: ATE > 0. Two important questions related to H_0 are:

1. What measure of *test statistics* is likely to be most informative to detect the departure from H_0?
2. How can we infer the distribution of the selected test statistic under H_0 and calculate the corresponding p-value?

As for the first question, the *test statistics* is a value computed from sample data in hypothesis testing; it refers to ATE estimate $\hat{\tau}$ in network A/B testing. Five test statistics were discussed in Section 2.3, including $\hat{\tau}_{naive}$, $\hat{\tau}_{neighbor}$, $\hat{\tau}_{Hajek}$, $\hat{\tau}_{lm_1}$, and $\hat{\tau}_{lm_2}$. These five test statistics can be classified into three types. The first type is simply mean difference between two groups, which is the naive estimator $\hat{\tau}_{naive}$. The second type of estimators $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{Hajek}$ consider users in effective treatment by only including users who are fractional q-neighborhood exposed to treatment/control given a fixed threshold q; then the difference between *effective groups* was measured. Specifically, $\hat{\tau}_{Hajek}$ further weighs users via their corresponding exposure probabilities and hence corrects the allocation bias. The third type of estimators include $\hat{\tau}_{lm_1}$ and $\hat{\tau}_{lm_2}$, measuring the difference by adding the level of exposure into the model. These estimators' performance with respect to both bias and variance were demonstrated and compared against each other in previous research work (Eckles et al., 2014; Gui et al., 2015). In this work, we focus on the second question about p-value calculation.

The analytical forms for distribution of any test statistic is not available due to interference in social network setting, thus the parametric test under H_0 is not feasible. An alternative way to test H_0 is using nonparametric test based on permutation (Jiang et al., 2016). The permutation test was based on repeated and random reassignment of each user's assignment z_i to treatment/control group. Let N^A and N^B denote the sizes for group A and group B from the original assignment. The permutation test reassigns users to two groups of size N^A and N^B

Algorithm 1: Permutation test conducts at individual level.

Input: graph G, assignment vector $\boldsymbol{Z}$, response vector $\boldsymbol{Y}$.
Compute test statistic $\hat{\tau}$ from the available sample $(G, \boldsymbol{Z}, \boldsymbol{Y})$.
for $m \in 1, \ldots, M$ **do**
  Randomly reassign $\boldsymbol{Z}$ to users, obtain new assignment vector $\boldsymbol{Z}^{(m)}$ with $\boldsymbol{Z}^{(m)} = (Z_1^{(m)}, Z_2^{(m)}, \ldots, Z_N^{(m)})$.
  Compute test statistic $\hat{\tau}^{(m)}$ from sample $(G, \boldsymbol{Z}^{(m)}, \boldsymbol{Y})$.
end
An approximate p-value is given by $M^{-1} \sum_{m=1}^{M} \mathbf{1}\{\hat{\tau}^{(m)} \geqslant \hat{\tau}\}$.

randomly and repeatedly, and test statistics $\hat{\tau}$ was estimated after each random reassignment. This process was repeated M times, then M estimates $\hat{\tau}_1, \hat{\tau}_2, \ldots, \hat{\tau}_M$ was obtained. They were ordered to construct the *permutation distribution* of test statistics $\hat{\tau}$ under H_0. Then the one-tailed p-value will be the number of entries in the permutation distribution greater than observed $\hat{\tau}$, divided by the total number of permutations M. This p-value is often called the *permutation p-value* and the statistical test is called a *permutation test* (Maris and Oostenveld, 2007). Jiang et al. (2016) used permutation test to calculate p-value for test statistics estimated from their proposed Ising model. This permutation test conducts at individual level, and the procedure is shown in Algorithm 1.

Now let us discuss how this permutation test at individual level will work for the three types of test statistics mentioned above. To make it clear, all test statistics follow the design of experiments based on graph cluster randomization. Under permutation at individual level, the assignment vector will be reassigned to users randomly. In other words, the size of treatment/control group remains the same after reassignment, but users in treatment/control group are completely randomly selected. Thus, the level of exposure σ_i for a user i is completely random after permutation. Which type of test statistics will get affected by this? The first type $\hat{\tau}_{naive}$ tends to be not sensitive since it ignores the interference among users completely. The second and third types are potentially more sensitive to permutation at individual level due to the change of level of exposure for users after permutation. Under the experimental design based on graph cluster randomization, users connected to each other tend to be put into the same cluster such that they may behave approximately the same as if they were in global treatment of interest. Yet permutations at individual level ignores this network structure. Thus ATE estimate $\hat{\tau}^{(m)}$ $(m = 1, \ldots, M)$ under permutation at individual level will be expected to behave very differently compared to $\hat{\tau}$ under graph cluster randomization, especially when strong local clustering exists. Note that the above discussion about sensitivity of three types of test statistics is from the significance test point of view, rather than test statistics's own performance with respect to bias and variance.

This issue about level of exposure can be largely circumvented by conducting the permutation at cluster level. Unlike permutations at individual level where users are correlated to each other due to the interference in social network, clusters can be regarded as *effectively independent*. The stronger the local clustering is, the closer the clusters get to independence. Let N users in the network be partitioned into S clusters $C_1, C_2, \ldots, C_S$. Let $\bar{Z}_{C_s}$ denote the assignment for cluster C_s $(s = 1, 2, \ldots, S)$. Let $C^{(i)}$ denote the cluster containing user i, then user i's assignment is determined by cluster $C^{(i)}$'s assignment. We have $Z_i = \bar{Z}_{C^{(i)}}$ $(i = 1, \ldots, N)$. Given the clusters $C_1, C_2, \ldots, C_S$, let $\bar{\boldsymbol{Z}}_{\boldsymbol{C}} = \{\bar{Z}_{C_1}, \bar{Z}_{C_2}, \ldots, \bar{Z}_{C_S}\}$ denote the assignment vector at cluster level. The assignment of user i will be the same as the assignment of the cluster $C^{(i)}$

Algorithm 2: Permutation test conducts at cluster level.

Input: graph G, assignment vector $\boldsymbol{Z}$, response vector $\boldsymbol{Y}$, partitioned S clusters $C_1, C_2, \ldots, C_S$.
Compute test statistic $\hat{\tau}$ from the available sample $(G, \boldsymbol{Z}, \boldsymbol{Y})$.
for $m \in 1, \ldots, M$ **do**
  Randomly reassign $\bar{\boldsymbol{Z}}_{\boldsymbol{C}}$ to clusters and obtain new assignment vector $\bar{\boldsymbol{Z}}_{\boldsymbol{C}}^{(\boldsymbol{m})} = \{\bar{Z}_{C_1}^{(m)}, \bar{Z}_{C_2}^{(m)}, \ldots, \bar{Z}_{C_S}^{(m)}\}$. Equivalently, $\boldsymbol{Z}^{(m)}$ is known. Compute test statistic $\hat{\tau}^{(m)}$ from sample $(G, \boldsymbol{Z}^{(m)}, \boldsymbol{Y})$.
end
An approximate p-value is given by $M^{-1} \sum_{m=1}^{M} \mathbf{1}\{\hat{\tau}^{(m)} \geqslant \hat{\tau}\}$.

that contains user i. Reassigning $\boldsymbol{Z}$ at cluster level is equivalent to reassigning $\bar{\boldsymbol{Z}}_{\boldsymbol{C}}$ directly. Also, knowing $\boldsymbol{Z}$ is equivalent to knowing $\bar{\boldsymbol{Z}}_{\boldsymbol{C}}$, because we have $Z_i = \bar{Z}_{C^{(i)}}$ for each individual user i $(i = 1, 2, \ldots, N)$. The procedure for permutation test at cluster level is shown in Algorithm 2.

Note that the accuracy of p-value increases with the number of draws M from the permutation distribution. The larger M, the more accurate the permutation p-value will be. In this work, the permutation p-value was calculated on $M = 1000$ repetitions.

4 Simulation Study

To investigate the performance of our proposed cluster-level permutation test (Ω_C) compared to the individual-level permutation test (Ω_I), a simulation study was conducted. This simulation study contains the complete process of experimentation in network A/B test study, including graph generation $G(V, E)$, treatment assignment $\boldsymbol{Z}$, and response values generation $\boldsymbol{Y}$, ATE estimation and p-value calculation. The network $G(V, E)$ was generated from small-world models (Watts and Strogatz, 1998) with N vertices and rewiring probability p_{rw}. The initial degree parameter d was fixed at $d = 10$. The graph was partitioned into S clusters $C_1, C_2, \ldots, C_S$ using balanced graph partitioning, and each cluster was randomly assigned to treatment/control group. Here S was fixed at $S = 10$. Given $\boldsymbol{Z} = \{Z_1, Z_2, \ldots, Z_N\} = \{\bar{Z}_{C^{(1)}}, \bar{Z}_{C^{(2)}}, \ldots, \bar{Z}_{C^{(N)}}\}$, the observed user response was generated based on a model of a stochastic function of the mean of neighbors' prior behaviors (Eckles et al., 2014) as follows:

$$
\begin{aligned}
Y_{i,t}^{*} &= \lambda_0 + \lambda_1 Z_i + \lambda_2 \frac{\boldsymbol{A}_{i,} \boldsymbol{Y}_{t-1}}{k_i} + \epsilon_{i,t}, \\
Y_{i,t} &= \mathbf{1}\left\{Y_{i,t}^{*} > 0\right\},
\end{aligned}
$$

where $Y_{i,t}$ is the observed response for user i at step t $(t = 0, 1, 2, \ldots)$, $\boldsymbol{Y}_t = (Y_{1,t}, Y_{2,t}, \ldots, Y_{N,t})^{\top}$ ($\top$ denotes transpose), k_i denotes degree of vertex i, and vector $\boldsymbol{A}_{i,}$ is the i-th row of adjacency matrix $\boldsymbol{A}$. Here $Y_{i,t}$ is a binary response value. Observed response values for all users were initialized at 0 when $t = 0$. Here λ_1 and λ_2 determines the strength of the direct treatment effect and network effect respectively. This process runs until a maximum time T is reached. Eckles et al. (2014) ran simulations with both $T = 3$ and $T = 10$ and get similar results. Thus in this work we set $T = 3$.

The factors of our simulation study are as follows: the randomness of edges (p_{rw}), the treatment effect level (λ_1), the network effect level (λ_2), and the graph size (N). As in Eckles et al. (2014), we varied the rewiring probability $p_{rw} \in \{0.00, 0.01, 0.10, 0.50, 1.00\}$, where $p_{rw} = 0.00$ corresponds to regular ring lattice and $p_{rw} = 1.00$ corresponds to graph with all random edges

respectively. The bigger p is, the more random edges and less clustering in the graph. We considered 5 levels for treatment effect $\lambda_1 \in \{0.0, 0.1, 0.2, 0.5, 1.0\}$ and 4 levels for network effect $\lambda_2 \in \{0.0, 0.5, 1.0\}$. We added λ_1 value at 0.1 and 0.2 in addition to values in Eckles et al. (2014), in order to evaluate the sensitivity of tests at weak treatment effect. The size of graph N was taken in $\{N_1, N_2, N_3\} = \{1000, 2000, 4000\}$. The threshold q in estimating $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{Hajek}$ was chosen to be 0.7, and λ_0 was fixed at -1.5.

For each scenario, a graph was first generated from the small-world models with chosen p_{rw}, then B data sets were generated. The bootstrap sample size B was set to 500 and all tests were carried out at a significance level of 0.05. For each bootstrap sample, we computed all five test statistics and calculated p-values for each test statistic under both permutation tests Ω_c and Ω_I, respectively. Here the number of permutation replicates is $M = 1000$. The performance of the tests was investigated in all scenarios. As discussed in section 2.3, the following test statistics were considered in the simulations: $\hat{\tau}_{naive}$, $\hat{\tau}_{neighbor}$, $\hat{\tau}_{Hajek}$, $\hat{\tau}_{lm_1}$, and $\hat{\tau}_{lm_2}$. Being classified into three types as discussed in Section 3, estimators within each type perform similarly based on our simulation results, as desired. Thus we will focus on the performance of tests for $\hat{\tau}_{naive}$, $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$ as representatives for each type of estimators. Next we will discuss the performance of tests Ω_C and Ω_I in two scenarios: size study and power study, corresponding to when H_0 is true and H_0 is false respectively.

4.1 Size Study

A type I error occurs if we reject the null hypothesis when the null hypothesis is true. Under the null hypothesis, the treatment effect $\lambda_1 = 0$, means that assigning users to either the control or treatment group will expose them to exactly the same experience. This is also equivalent to *A/A Test*, an experiment where users are assigned to one of two groups, but their experience is the same despite of the group assignment. Since there is no true difference between treatment and control group, all significant τs identified by the test should be false positive tests. Ideally, the null hypothesis should be rejected 5% of the time.

As shown in Figure 1, test Ω_C holds its level reasonably well for all estimators. When $\lambda_1 = 0.0$, the percent of positive tests observed by Ω_C closely corresponds to 0.05, proving that this test controls false positive rate. Test Ω_I holds its level well only for estimator $\hat{\tau}_{naive}$. For $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$, Ω_I produces 0% false positive rates; A further observation of histogram for p-values shows that p-values all concentrate over a small interval, rather than uniformly distributed over [0, 1] interval.

4.2 Power Study

The statistical power is the probability of rejecting the null hypothesis when the null hypothesis is false – that is, the ability of a test to detect an effect if the effect actually exists. Here H_0 is false when λ_1 is greater than zero.

We first assess the power of two permutation tests Ω_C and Ω_I for $\hat{\tau}_{naive}$, $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$ when fixing p_{rw} at $p_{rw} = 0.0$ (Figure 1). From Figure 1, three estimators shows similar trend of performance for Ω_C (represented by solid lines in Figure 1). For all three estimators, the test power increases as treatment effect λ_1 increases. The test power also increases as the number of users N gets larger. At the smallest treatment effect $\lambda_1 = 0.1$, Ω_C can detect the effect at least 40% of times at $N = N_3 = 4000$. Increasing λ_1 to 0.2, the power reaches to 60% or 70% of times even at $N = N_2 = 4000$; it reaches almost 100% for $\lambda_1 = 0.5$ even with the smallest

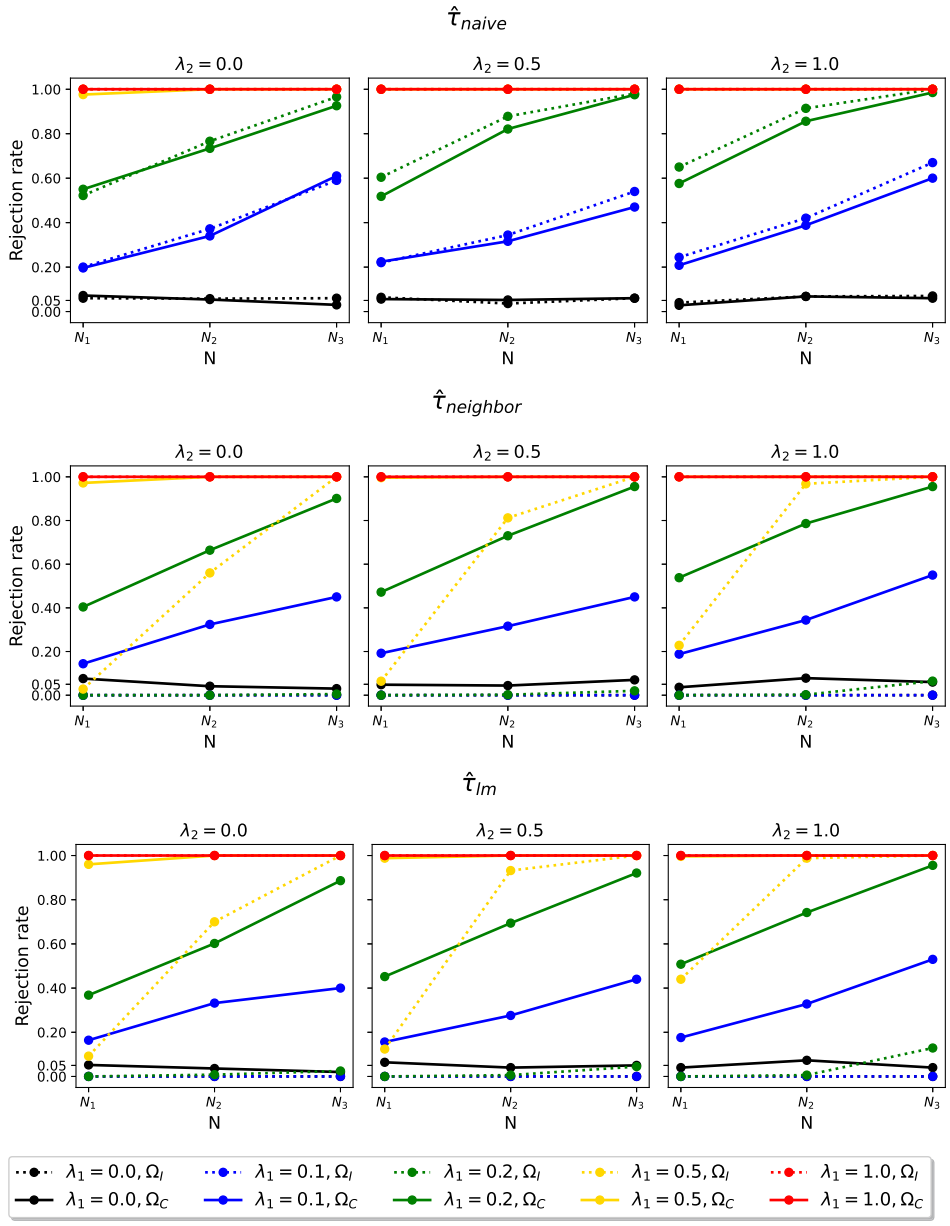

Figure 1: Rate of rejection of H_0 for $\hat{\tau}_{naive}$ (top panel), $\hat{\tau}_{neighbor}$ (middle panel) and $\hat{\tau}_{lm_1}$ (bottom panel) at different combinations of λ_1, λ_2 and N, with fixing $p_{rw} = 0.0$.

$N = N_1 = 1000$. And, our test Ω_C is quite robust to the change of network effect λ_2. Comparing the power of Ω_C at $\lambda_2 \in \{0.0, 0.5, 1.0\}$ in Figure 1, we even see a slightly increasing power as λ_2 increases. This may come from our data generation mechanism of adding up treatment effect and network effect; strong network effect will strengthen user's treatment effect and thus makes the detect of treatment effect slightly easier. In contrast, the test Ω_I behaves quite differently for $\hat{\tau}_{naive}$, $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$. For $\hat{\tau}_{naive}$, Ω_I behaves similarly as Ω_C as we expected due to its insensitivity to network exposure. For $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$, Ω_I performs poorly; it fails to detect the effect, with 0% rejection rate when λ_1 is at 0.1 and 0.2 even at $N = N_5 = 5000$. In the case of

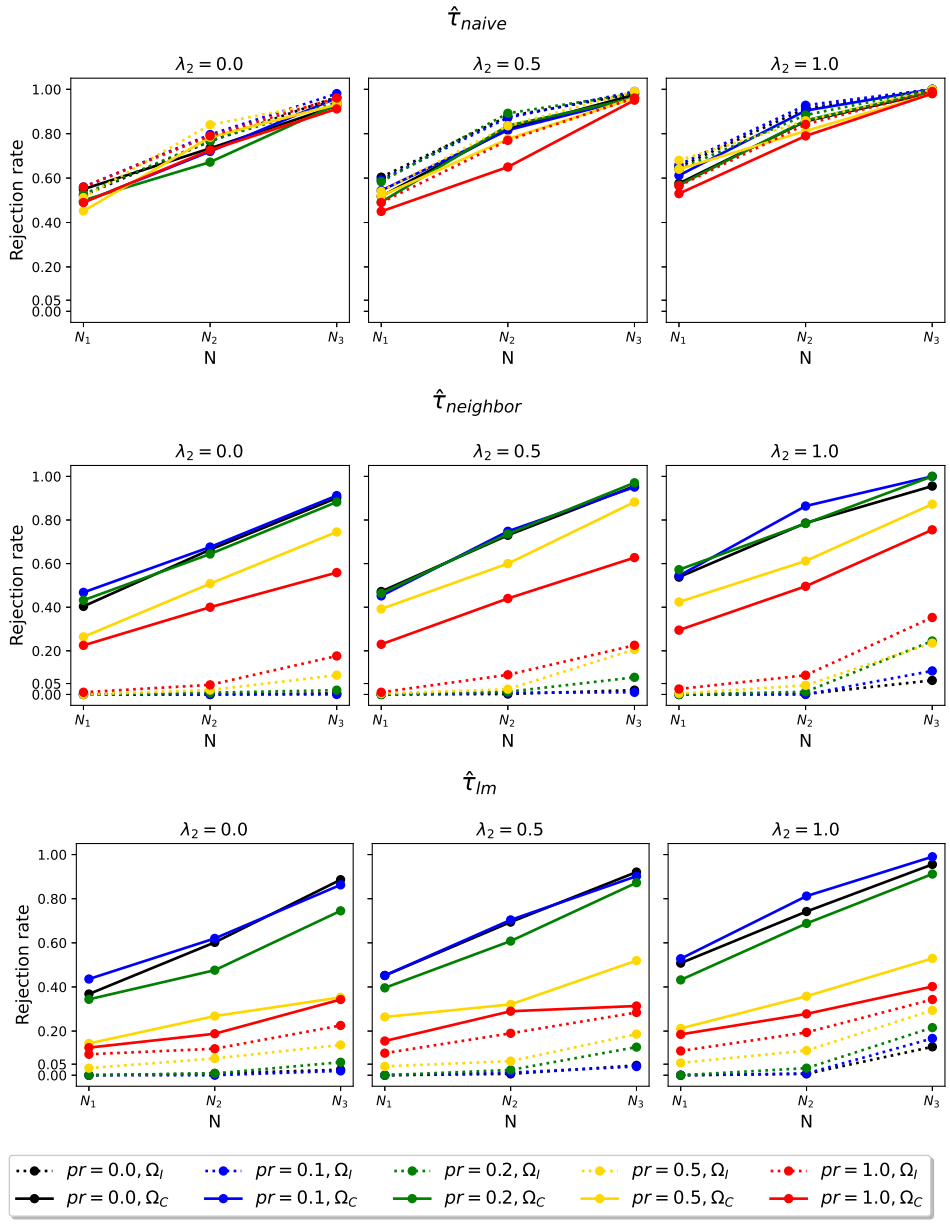

Figure 2: Rate of rejection of H_0 for $\hat{\tau}_{naive}$ (top panel), $\hat{\tau}_{neighbor}$ (middle panel) and $\hat{\tau}_{lm_1}$ (bottom panel) at different combinations of p_{rw}, λ_2 and N, with fixing $\lambda_1 = 0.2$.

$\lambda_2 = 0.0$, one may question why Ω_I performs poorly for $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$ even without network effect. Note that, unlike $\hat{\tau}_{naive}$, $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$ are computed by considering the factor of level of exposure; Ω_I breaks the clustering at experimental design and reassigns assignments to users randomly, leading to poor performance.

We now look at the performance of cluster-level permutation test Ω_C vs individual-level permutation test Ω_I with varying rewiring probability p_{rw} (see Figure 2). For $\hat{\tau}_{naive}$, tests Ω_C and Ω_I behave similarly with varying p_{rw}, since $\hat{\tau}_{naive}$ is not sensitive to graph structure. Two tests behave very differently for $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$. The power of test Ω_C decreases as p_{rw} increases,

with a big drop when p_{rw} increases from 0.1 to 0.5. With $p_{rw} = 0.5$, many random edges and less clustering in the graph makes clusters in G harder to get close to independent events. As for the test Ω_I, it fails to detect the effect, with 0% power when p_{rw} is at 0.01 and 0.1. Interestingly, the power of Ω_I goes up slightly for $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$ as p_{rw} increases from 0.1 to 0.5, with opposite direction compared to Ω_C. For $\hat{\tau}_{lm_1}$, even at $p_{rw} = 1.0$, which corresponds to Ω_C's worst scenario and Ω_I's best scenario, Ω_C still performs better than Ω_I. Overall, the power of test Ω_C is larger when the network has more local clustering. If the network has little local clustering, then benefits of conducting permutation at cluster level are reduced.

5 Discussion

In this work, we described the entire procedure for designing experiment, estimating ATE, and testing the null hypothesis of no ATE between two groups in the network A/B testing framework. Due to the violation of SUTVA in social network settings, analytical forms for distributions under the null hypothesis are not available for test statistics. It is natural to perform nonparametric tests based on permutation. Permutation has the advantage of being able to be used for testing any statistics of interest if events can be assumed to be independent, regardless of its theoretical tractability under the null hypothesis. In contrast to the work in Jiang et al. (2016) who used a permutation test at user level, we proposed a novel method by permutation of users' assignment at cluster level, which can make users' level of neighborhood exposure remain similar as they were in the original experimental design to the largest extent. Our cluster-level permutation test considers the graph structure for each permutation and minimizes the "contamination" across clusters in contrast to the individual-level permutation test. Our work also compared two permutation tests by testing three different types of test statistics for estimating ATE. Specifically, we have estimated all five test statistics and tested each of these against both a null distribution derived from repeated permutation of assignment vector at cluster level and a null distribution derived from repeated permutation at the individual level. To the best of our knowledge, this is the first study to propose a nonparametric permutation test considering the graph structure in network A/B testing framework. It also represents the first comparative appraisal of permutation tests at the cluster level vs at the individual level in network A/B testing via intensive simulation study for different types of test statistics.

The simulation study shows that our proposed test Ω_C performs well. Nominal type I error controls for all three types of test statistics. When treatment effect exists, Ω_C can detect with high power in most cases. In general, the power of Ω_C increases as treatment effect λ_1 increases and the graph size N increases. The performance of test Ω_C is robust to the strength of the network effect. Note that for $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$, the power of Ω_C drops when p_{rw} increases from 0.1 to 0.5. This can be explained by the fact that a large rewiring probability p_{rw} indicates there are many random edges and less clustering in the graph and thus clusters in the graph can not be regarded as effectively independent. Thus sampling is an important factor for determining the power of our test, which urges practitioners to pay attention to the local clustering of graph G when sampling G from the entire network. Our proposed test Ω_C is superior to Ω_I for $\hat{\tau}_{neighbor}$ and $\hat{\tau}_{lm_1}$ in all scenarios, while Ω_C performs equivalently to Ω_I for estimator $\hat{\tau}_{naive}$ in that $\hat{\tau}_{naive}$ is not sensitive to network exposure. Overall, this cluster-level permutation test is an effective and easily implemented approach for testing the null hypothesis of ATE = 0, ascertaining the distribution under null hypothesis, and deriving the corresponding p-value in the network A/B testing framework.

Supplementary Material

The zip supplementary material file contains the Python scripts for generating graph data, computing ATE estimators, estimating p-value via permutation tests, and generating figures in this paper.

Acknowledgement

We are grateful to Dr. Zhigeng Geng and Dr. Alyssa Glass for the inspiring discussions to make the paper better. We would like to thank the reviewers for their constructive suggestions to improve the paper. We would also like to thank Dr. Shaozhen Ma for the visualization improvement.

References

Aronow PM, Samii C (2012). Estimating average causal effects under general interference. In: *Summer Meeting of the Society for Political Methodology*, University of North Carolina, Chapel Hill, July, 19–21, Citeseer.

Backstrom L, Kleinberg J (2011). Network bucket testing. In: *Proceedings of the 20th International Conference on World Wide Web, WWW'11*, 615–624. ACM, New York, NY, USA.

Eckles D, Karrer B, Ugander J (2014). Design and analysis of experiments in networks: Reducing bias from interference. arXiv preprint: https://arxiv.org/abs/1404.7530.

Gui H, Xu Y, Bhasin A, Han J (2015). Network A/B testing: From sampling to estimation. In: *Proceedings of the 24th International Conference on World Wide Web, WWW'15*, 399–409. ACM, New York, NY, USA.

Gupta A, Krauthgamer R, Lee JR (2003). Bounded geometries, fractals, and low-distortion embeddings. In: *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, 534–543. IEEE.

Jiang B, Shi X, Shang H, Geng Z, Glass A (2016). *A Framework for Network A/B Test*. arXiv preprint: https://arxiv.org/abs/1610.07670.

Karrer B, Shi L, Bhole M, Goldman M, Palmer T, Gelman C, et al. (2021). Network experimentation at scale. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3106–3116.

Karypis G, Kumar V (1998). Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48(1): 96–129. https://doi.org/10.1006/jpdc.1997.1404

Katzir L, Liberty E, Somekh O (2012). Framework and algorithms for network bucket testing. In: *Proceedings of the 21st International Conference on World Wide Web, WWW'12*, 1029–1036. ACM, New York, NY, USA.

Kohavi R, Deng A, Frasca B, Longbotham R, Walker T, Xu Y (2012). Trustworthy online controlled experiments: Five puzzling outcomes explained. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 786–794. ACM.

Kohavi R, Deng A, Frasca B, Walker T, Xu Y, Pohlmann N (2013). Online controlled experiments at large scale. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1168–1176.

Kohavi R, Deng A, Longbotham R, Xu Y (2014). Seven rules of thumb for web site experimenters. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'14*, 1857–1866. ACM, New York, NY, USA.

Kohavi R, Longbotham R, Walker T (2010). Online experiments: Practical lessons. *Computer*, 43(9): 82–85. https://doi.org/10.1109/MC.2010.264

Kohavi R, Tang D, Xu Y (2020). *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing*. Cambridge University Press.

Liu Y, Zhou Y, Li P, Hu F (2022). Adaptive A/B test on networks with cluster structures. In: *International Conference on Artificial Intelligence and Statistics*, 10836–10851. PMLR.

Maris E, Oostenveld R (2007). Nonparametric statistical testing of EEG-and MEG-data. *Journal of Neuroscience Methods*, 164(1): 177–190. https://doi.org/10.1016/j.jneumeth.2007.03.024

Nishimura J, Ugander J (2013). Restreaming graph partitioning: simple versatile algorithms for advanced balancing. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1106–1114.

Rubin DB (1986). Comment: Which ifs have causal answers? *Journal of the American Statistical Association*, 81(396): 961–962.

Saveski M, Pouget-Abadie J, Saint-Jacques G, Duan W, Ghosh S, Xu Y, et al. (2017). Detecting network effects: Randomizing over randomized experiments. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1027–1035.

Ugander J, Backstrom L (2013). Balanced label propagation for partitioning massive graphs. In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, 507–516.

Ugander J, Karrer B, Backstrom L, Kleinberg J (2013). Graph cluster randomization: Network exposure to multiple universes. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'13*, 329–337. ACM, New York, NY, USA.

Ugander J, Yin H (2023). Randomized graph cluster randomization. *Journal of Causal Inference*, 11(1): 20220014. https://doi.org/10.1515/jci-2022-0014

Watts DJ, Strogatz SH (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684): 440–442. https://doi.org/10.1038/30918