Detecting Interference in A/B Testing with Increasing Allocation

Kevin Han[1], Shuangning Li[2], Jialiang Mao[3], and Han Wu[1]

[1]Department of Statistics, Stanford University, CA, USA
[2]Department of Statistics, Harvard University, MA, USA
[3]LinkedIn Corporation, Sunnyvale, CA, USA

March 27, 2023

Abstract

In the past decade, the technology industry has adopted online randomized controlled experiments (a.k.a. A/B testing) to guide product development and make business decisions. In practice, A/B tests are often implemented with increasing treatment allocation: the new treatment is gradually released to an increasing number of units through a sequence of randomized experiments. In scenarios such as experimenting in a social network setting or in a bipartite online marketplace, interference among units may exist, which can harm the validity of simple inference procedures. In this work, we introduce a widely applicable procedure to test for interference in A/B testing with increasing allocation. Our procedure can be implemented on top of an existing A/B testing platform with a separate flow and does not require *a priori* a specific interference mechanism. In particular, we introduce two permutation tests that are valid under different assumptions. Firstly, we introduce a general statistical test for interference requiring no additional assumption. Secondly, we introduce a testing procedure that is valid under a time fixed effect assumption. The testing procedure is of very low computational complexity, it is powerful, and it formalizes a heuristic algorithm implemented already in industry. We demonstrate the performance of the proposed testing procedure through simulations on synthetic data. Finally, we discuss one application at LinkedIn, where a screening step is implemented to detect potential interference in all their marketplace experiments with the proposed methods in the paper.

Keywords: causal inference, SUTVA, online experiments, hypothesis testing, permutation test.

1 Introduction

The technology industry has adopted online randomized controlled experiments, also known as A/B testing, to guide product development and make business decisions [Kohavi et al., 2013, 2020]. In the past decade, firms have developed a dynamic phase release framework in which a new treatment (such as a new product feature) is gradually released to an increasing number of units in the target population through a sequence of randomized experiments [Kohavi et al., 2020]. Companies including Google, Microsoft, LinkedIn, and Meta all developed in-house platforms that implement this framework at-scale [Tang et al., 2010, Kohavi et al., 2013, Bakshy et al., 2014, Xu et al., 2015]. Contrary to the sophisticated engineering design of such platforms, the strategy to analyze A/B testing is relatively simple—often, only the most powerful experiment in the sequence is used to provide a summary of the treatment effect, using tools from classical causal inference assuming independence among test units [Imbens and Rubin, 2015].

In scenarios such as experimenting in a social network setting or in a bipartite online marketplace, interference among units may exist. Thus, a natural question is whether such interference harms the validity of simple inference procedures. Specific designs have been proposed to test or correct for the interference effects in different applications [Saveski et al., 2017, Eckles et al., 2017, Ugander et al., 2013, Pouget-Abadie et al., 2019a, Johari et al., 2022]. However, these designs are limited to specific applications and often require significant engineering work to implement in parallel to the existing A/B testing infrastructure in most companies. Even when such designs are implemented, their complex nature often results in lower throughput and can slow down the decision process.

In this work, we introduce a widely applicable procedure to test for interference in generic online experiments. The proposed method utilizes data from multiple experiments in the sequence. It can be implemented on top of an existing A/B testing platform with a separate flow and does not require *a priori* the knowledge of the underlying interference mechanism. Once implemented, this test can be run as a standard screening for any A/B test running on the platform. If the test suggests that no interference exists, the experimenter can proceed with classical causal inference analysis with confidence; if the test suggests that some form of interference does exist, the experimenter may need to redesign experiments in a more delicate way. At the platform level, such screening could provide valuable and timely feedback on the choice of designs and help experimenters update development roadmaps accordingly.

1.1   A motivating example and our contribution

The most straightforward statistical analysis following A/B tests is to compute the difference-in-means estimator, i.e., the difference in the average of outcomes of the treatment group and that of the control group. Under the classical Stable Unit Treatment Value Assumption (SUTVA), which requires that the potential outcomes for any unit do not vary with the treatments assigned to other units, one can easily show that the difference-in-means estimator will be close to the causal effect as long as the sample size is large [Imbens and Rubin, 2015]. This implies that when we compute the difference-in-means estimator for any single randomized experiments in an A/B test with increasing allocation, the value of the estimator should not change by much. However, in some real-world scenarios, we observe drastic change in the difference-in-means estimators throughout the experiments. In Figure 1, we show an example from an A/B test implemented by LinkedIn. In this example, we see that the difference-in-means estimator decreases as the treatment is released to more units. We naturally wonder: What causes this phenomenon? Could it be purely due to randomness? Is the SUTVA assumption violated in this case?

One plausible explanation for this phenomenon is the existence of interference, i.e., when treatment assigned to one unit may affect observed outcomes for other units. One form of interference is marketplace competition. Imagine a new treatment that can help units perform better in the market. For any particular unit, the treatment brings benefit, but when more of the other units are treated, the other units become more competitive and thus negatively impact the performance of that particular unit. Therefore, in these cases, we often observe that the difference-in-means estimator decreases with treatment probability. Indeed, the experiments in Figure 1 were run in a setting with marketplace competition. One other common form of interference is through social networks. People's behaviors tend to be positively correlated with those of others connected to them in the network. Think about a treatment that encourages users to comment on a social media platform: users tend to comment more when they see comments from friends. In these cases, we usually observe the difference-in-means estimator increasing with treatment probability.
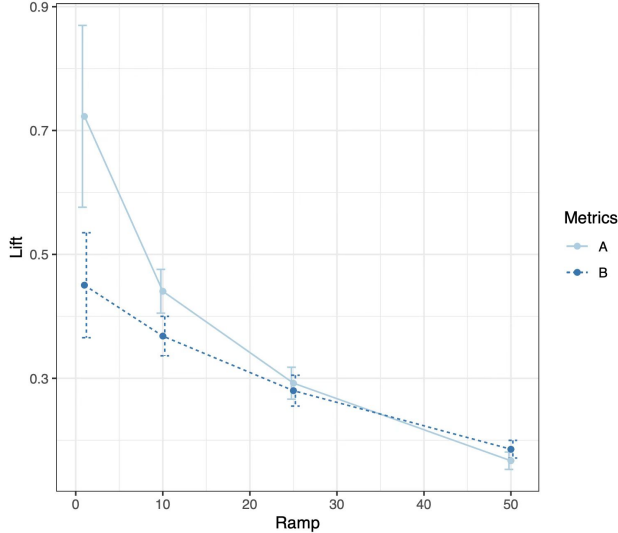
Figure 1: An A/B test implemented by LinkedIn with increasing allocation. On the x-axis, we show the percentage of units that are in the treatment group; on the y-axis, we show the value of the difference-in-means estimator. Note that A and B stand for different outcome metrics.

In practice, however, the structure of interference can be more complicated than the two apparent forms discussed in the above paragraph. Often, experimenters manually examine the difference-in-means plot and decide whether to send the job to other experimentation platforms that deal with interference more carefully. We need a way to formally test whether interference exists.

In this work, we introduce statistical testing procedures that test for interference in A/B testing with increasing allocation. The methods we propose are scalable and parallelable. They are also agnostic to interference mechanism: even if we have no knowledge of the interference structure, the testing procedure is still valid. Knowledge of the interference structure can, however, be helpful in increasing the power of the testing procedure. We introduce two different testing strategies under different assumptions in Sections 3.1 and 3.2. In Section 3.1, we introduce a general statistical test for interference, a test that requires no additional assumptions. The proposed method is inspired by the testing procedure proposed by Athey et al. [2018], but it is more powerful than that of Athey et al. [2018] by making use of multiple experiments. In Section 3.2, we introduce a testing procedure that is valid under a time fixed effect assumption. The testing procedure is of very low computational complexity, and it is more powerful than the test proposed in Section 3.1. In particular, one special case of this method formalizes a heuristic algorithm discussed above, which decides that interference exists when the difference-in-means estimators are very different.

1.2   Related work

The classical literature on causal inference often assumes that there is no cross-unit interference. When interference presents, many classical inference methods break down. Interest in causal inference with interference started in the social and medical sciences [Sobel, 2006, Hudgens and Halloran, 2008]. Since then, one line of work focuses on estimation and inference of treatment effects under network interference [Tchetgen and VanderWeele, 2012, Toulis and Kao, 2013, Aronow and Samii, 2017, Sussman and Airoldi, 2017, Basse and Feller, 2018, Bhattacharya et al., 2020, Leung, 2020, Sävje et al., 2021, Sävje, 2021, Hu et al., 2022, Li and Wager, 2022]. In order to facilitate estimation,

these works either assume that there are special randomization designs or that the interference has some restricted form defined by a given network. Applications to A/B testing are also considered in Ugander et al. [2013], Eckles et al. [2017], and Basse and Airoldi [2018]. One assumption implicitly made in these works is that the experiment is conducted only once. In the multiple experiments regime, Viviano [2020] studies the design of two-wave experiments under interference. Yu et al. [2022] and Cortez et al. [2022] consider estimating the total treatment effects under interference with data from more than two time steps. Bojinov et al. [2021] and Han et al. [2021] further investigate the problem in panel experiments. Our work differs from the above works for at least two reasons: (1) instead of focusing on estimation, we focus on testing whether interference exists and (2) we do not need to make additional assumptions in order for the testing procedure to be valid.

In the literature of testing for interference, Bowers et al. [2013] consider model-based approaches, Pouget-Abadie et al. [2019b] introduce an experimental design strategy, and Aronow [2012] and Athey et al. [2018] propose conditional randomization tests restricted to a subset of what they call focal units, and a subset of assignments that make the null hypothesis sharp for focal units. Basse et al. [2019] and Puelz et al. [2022] further extend this method by using a conditioning mechanism to allow the selection of focal units to depend on the observed treatment assignment. However, none of these works addresses the problem of multiple experiments, and their methods tend to have lower power when directly applied in our setup. To the best of our knowledge, our work is the first to consider testing interference with a sequence of randomized experiments.

Our work is also related to research on interference in online marketplace experiments (See Basse et al. [2016], Fradkin [2019], Holtz et al. [2020], Bajari et al. [2021], Wager and Xu [2021], Johari et al. [2022], Li et al. [2022] among others). This line of work usually requires careful modeling of the market and the interference mechanism. The testing procedure introduced in this paper, in contrast, can be applied to arbitrary forms of interference.

2   Problem Setup

We work in a setting where we run a sequence of A/B tests with increasing allocations. Formally, suppose that there are K experiments on a population of n units. Let π_k be the marginal treatment probability of the kth experiment. The treatment probabilities satisfy $\pi_1 < \pi_2 < \cdots < \pi_K$. For each experiment $k \in \{1, \ldots, K\}$ and each unit $i \in \{1, \ldots, n\}$, let

$$W_{i,k} := \text{treatment of unit } i \text{ assigned in the } k^{\text{th}} \text{ experiment,}$$

$$Y_{i,k} := \text{outcome of unit } i \text{ in the } k^{\text{th}} \text{ experiment.}$$

Here we assume that $W_{i,k} \in \{0, 1\}$ is a binary treatment variable and that a value of 1 corresponds to the treatment group while a value of 0 corresponds to the control group.

The experiments are implemented in the following way. In the first experiment, each unit i is randomly assigned a treatment $W_{i,1}$, where

$$W_{i,1} \sim \text{Bernoulli}(\pi_1) \text{ independently.} \tag{1}$$

In the subsequent experiments, more units are assigned to the treatment group. Specifically, conditioning on the previous treatments, each $W_{i,k}$ is sampled from the following distribution independently:

$$\begin{cases} W_{i,k} \sim \text{Bernoulli}\left((\pi_k - \pi_{k-1})/(1 - \pi_{k-1})\right), & \text{if } W_{i,k-1} = 0; \\ W_{i,k} = 1, & \text{if } W_{i,k-1} = 1. \end{cases} \tag{2}$$

4

This formulation guarantees that if we look at the kth experiment alone, then the treatments $W_{i,k}$'s are i.i.d. Bernoulli(π_k).

Let $W_{1:n,1:K}$ be the $n \times K$ treatment matrix and $Y_{1:n,1:K}$ be the $n \times K$ outcome matrix of all units and all experiments. Let $X_i \in \mathbb{R}^d$ be the observed covariates of unit i that do not change over the course of the experiments. Correspondingly, let $X_{1:n} \in \mathbb{R}^{n \times d}$ be the matrix of covariates of all units.

Following the Neyman-Rubin causal model, we assume that potential outcomes $Y_{i,k}(w_{1:n,1:K}) \in \mathbb{R}$ exist for all $w_{1:n,1:K} \in \{0,1\}^{n \times K}$ and that the observed outcomes satisfy $Y_{i,k} = Y_{i,k}(W_{1:n,1:K})$.[1] The goal is to test the following hypothesis:

Hypothesis 1 (No cross-unit interference)**.** $Y_{i,k}(w_{1:n,1:K}) = Y_{i,k}(\tilde{w}_{1:n,1:K})$ if $w_{i,1:K} = \tilde{w}_{i,1:K}$.

The hypothesis states that the outcomes of unit i depend only on the treatments of unit i and not on the treatments of others. We call this hypothesis the no cross-unit interference hypothesis.

3 Testing for interference

In this section, we introduce methods that test for the existence of cross-unit interference. For brevity's sake, we focus on testing with two experiments. We then discuss further extensions to multiple experiments in Section 3.5.

Naturally, the first question that occurs is how interference might arise. To formalize this, we introduce a notion of *candidate exposure* that captures the potential form of interference. Using domain knowledge, experimenters can specify the candidate exposure, which can vary from application to application. When we consider user-level data, we have a natural social network. Here experimenters may suspect that a user's outcome is influenced by treatments of "friends", i.e., users connected through the social network. And thus in this example, some plausible choices of candidate exposures include the fraction of friends who are treated, and the number of friends who are treated. When we consider marketplace competition, advertisers are the subjects of treatment. Here, the sales of an advertiser may be impacted by the treatments of competitors, i.e., advertisers that sell similar products. Hence, in this application, experimenters can choose candidate exposures to be the number of treated advertisers that sell products of the same category, or an average of treatments given to other advertisers weighted by some product similarity metric.

Formally, for each experiment k and each unit i, we use $H_{i,k} = h_i(W_{-i,k}) \in \mathbb{R}^m$ to denote the candidate exposure. Here $W_{-i,k}$ is the treatments given to all other units except i in the kth experiment. We use the form $h_i(W_{-i,k})$ to emphasize that the candidate exposure depends on other units' treatments. We also write $H_{1:n,k} = (H_{1,k}, H_{2,k}, \ldots, H_{n,k})^\top \in \mathbb{R}^{n \times m}$ to reference the candidate exposures of all units.

We want to emphasize that for all the tests introduced below, we do not require the candidate exposure to be correctly specified in order for the tests to be valid. However, the form of the candidate exposure matters for the power of the tests.

We will then move on to test the hypothesis that no interference exists making use of the candidate

---

[1]In the literature, a *no anticipation effects* assumption is often made in such potential outcome models. The assumption states that the outcome $Y_{i,k}$ depends only on the treatments assigned during and prior to the kth experiment. With this assumption, the potential outcomes can be written as $Y_{i,k}(w_{1:n,1:k})$ which satisfies $Y_{i,k} = Y_{i,k}(W_{1:n,1:k})$. Here, for simplicity, we keep the original notation.

exposure $H_{i,k}$. In the following sections, we discuss different strategies to test for interference under different assumptions.

3.1 Testing under general assumptions

We start with a setting where we have access to a dataset from *only one* experiment. Suppose that we collect data on units indexed by $i = 1, ..., n$, where each unit is randomly assigned to a binary treatment $W_i \in \{0, 1\}$,

$$W_i \sim \text{Bernoulli}(\pi) \text{ independently} \tag{3}$$

for some $0 \le \pi \le 1$. For each unit, we observe an outcome of interest $Y_i \in \mathbb{R}$ and some covariates $X_i \in \mathbb{R}^p$. Athey et al. [2018] proposed a method to test for Hypothesis 1 in this setting.[2] We sketch the procedure in Algorithm 1.

Algorithm 1 Testing for interference effect (one experiment).

Input: Dataset $\mathcal{D} = (W_{1:n}, X_{1:n}, Y_{1:n}, H_{1:n})$, exposure function h, test statistic T.

1. Randomly split the data into two folds. Let $\mathcal{I}_{\text{foc}}$ and $\mathcal{I}_{\text{aux}}$ be the index set for the first fold (focal units) and the second fold (auxiliary units). Write the first fold of data as $\mathcal{D}_{\text{foc}} = (W_{\text{foc}}, X_{\text{foc}}, Y_{\text{foc}}, H_{\text{foc}})$ and the second as $\mathcal{D}_{\text{aux}} = (W_{\text{aux}}, X_{\text{aux}}, Y_{\text{aux}}, H_{\text{aux}})$.
2. Compute a test statistic $T^{(0)} = T(W_{\text{foc}}, X_{\text{foc}}, Y_{\text{foc}}, H_{\text{foc}})$ that captures the importance of H in predicting Y.
3. **For** $b = 1, \ldots B$:

    Regenerate treatments for the auxiliary units: $\widetilde{W}_i^{(b)} \sim \text{Bernoulli}(\pi)$ for $i \in \mathcal{I}_{\text{aux}}$.
    Recompute the candidate exposure for focal units: $\widetilde{H}_i^{(b)} = h_i(W_{\text{foc} \setminus \{i\}}, \widetilde{W}_{\text{aux}}^{(b)})$ for $i \in \mathcal{I}_{\text{foc}}$.
    Recompute the test statistic: $T^{(b)} = T(W_{\text{foc}}, X_{\text{foc}}, Y_{\text{foc}}, \widetilde{H}_{\text{foc}}^{(b)})$.

    End For

Output: The p-value

$$p = \frac{1}{B+1} \left(1 + \sum_{b=1}^{B} \mathbb{1} \left\{ T^{(0)} \le T^{(b)} \right\} \right). \tag{4}$$

Algorithm 1 requires as input a test statistic T that captures the importance of the candidate exposure H in predicting outcome Y. As an illustration, assume for now that $H_i \in \mathbb{R}$. One plausible choice of the test statistic T (when $H_i \in \mathbb{R}$) is the following: we run a linear regression of $Y_{\text{foc}} \sim W_{\text{foc}} + X_{\text{foc}} + H_{\text{foc}}$, extract the coefficient of H_{foc}, and take the test statistic T to be the absolute value of the coefficient. We use this regression coefficient statistic as an example to explain the intuition of the algorithm. Under the null hypothesis, the candidate exposure H has no power to predict the outcome Y before or after regenerating treatments, and thus the distribution of the test statistic T will not change after regenerating treatments. Hence, the p-value will be stochastically larger than Unif$[0, 1]$. Under the alternative hypothesis, the behavior of the p-value can be very different. Consider a simple example where H_i is the treatment assigned to the closest friend of unit i and $Y_i = \alpha^\top X_i + \beta W_i + \theta H_i + \epsilon_i$ for some i.i.d. zero mean errors ϵ_i. In this example, the original test statistic $T(W_{\text{foc}}, X_{\text{foc}}, Y_{\text{foc}}, H_{\text{foc}}) \approx |\theta|$ when the sample size is large. However, after regenerating treatments, for each focal unit i, if the closest friend of i is among the auxiliary units,

---

[2]The method proposed by Athey et al. [2018] is more general. Here, we focus on a special case: testing the existence of cross-unit interference in Bernoulli experiments.

Algorithm 2 Testing for interference effect (two experiments).

Input: Datasets $\mathcal{D}_1 = (W_{1:n,1}, X_{1:n}, Y_{1:n,1}, H_{1:n,1})$, $\mathcal{D}_2 = (W_{1:n,2}, X_{1:n}, Y_{1:n,2}, H_{1:n,2})$, exposure function h, test statistic T.

1. Let $\mathcal{I}_{\mathrm{nc}} = \{i : W_{i,1} = W_{i,2}\}$ be the set of units whose treatment didn't change over the experiments. Randomly sample a subset of $\mathcal{I}_{\mathrm{nc}}$ of size $n/2$. We call the subset $\mathcal{I}_{\mathrm{foc}}$. Let $\mathcal{I}_{\mathrm{aux}} = [n] \setminus \mathcal{I}_{\mathrm{foc}}$.
2. Take the difference of $Y_{\mathrm{foc},2}$ and $Y_{\mathrm{foc},1}$: let $Y_{\mathrm{foc}}^{\mathrm{diff}} = Y_{\mathrm{foc},2} - Y_{\mathrm{foc},1}$. Compute a test statistic $T^{(0)} = T(W_{\mathrm{foc},1:2}, X_{\mathrm{foc}}, Y_{\mathrm{foc}}^{\mathrm{diff}}, H_{\mathrm{foc},1:2})$ that captures the importance of H in predicting Y^{diff}.
3. **For** $b = 1, \ldots B$:

   Randomly permute treatments for the auxiliary units of the data: $\widetilde{W}_{i,1:2}^{(b)} = W_{\sigma^{(b)}(i),1:2}$ for $i \in \mathcal{I}_{\mathrm{aux}}$, for some permutation $\sigma^{(b)}$ of $\mathcal{I}_{\mathrm{aux}}$.

   Recompute the candidate exposure for the focal units: $\widetilde{H}_{i,k}^{(b)} = h_i(W_{\mathrm{foc} \setminus \{i\},k}, \widetilde{W}_{\mathrm{aux},k}^{(b)})$ for $i \in \mathcal{I}_{\mathrm{foc}}$ and $k \in \{1, 2\}$.

   Recompute the test statistic: $T^{(b)} = T(W_{\mathrm{foc},1:2}, X_{\mathrm{foc}}, Y_{\mathrm{foc}}^{\mathrm{diff}}, \widetilde{H}_{\mathrm{foc},1:2}^{(b)})$.

   End For

Output: The p-value

$$p = \frac{1}{B+1} \left(1 + \sum_{b=1}^{B} \mathbb{1}\left\{ T^{(0)} \leq T^{(b)} \right\} \right). \tag{5}$$

---

then $\widetilde{H}_i$ is marginally a Bern(π) random variable, *independent* of Y_i; and hence the distribution of $T(W_{\mathrm{foc}}, X_{\mathrm{foc}}, Y_{\mathrm{foc}}, \widetilde{H}_{\mathrm{foc}}^{(b)})$ will not concentrate around $|\theta|$. In this case, the p-value is far from the Unif$[0, 1]$ distribution.

In practice, experimenters can use any test statistic T that are suitable for specific applications. For example, if the covariate X is of high dimension, a lasso-type algorithm can be used. One can also run more complicated machine learning algorithms, e.g., random forest and gradient boosting, with Y as a response and X, W, H as predictors, and set the statistic T to be any feature importance statistic of H. Just like the choice of candidate exposure h, the choice of test statistic T will not hurt the validity of the test, but will largely influence the power of the test.

Then a natural question to ask is whether we can make use of information from multiple experiments to further increase the power of the test. Suppose that we collect data from *two* experiments on the same n units indexed by $i = 1, \ldots, n$. In order to increase the power of the previous testing procedure, a natural idea is to reduce the variance in the test statistic computed in Algorithm 1. To do so, instead of focusing on $Y_{i,2}$ itself, we focus on $Y_{i,2} - Y_{i,1}$. This difference is helpful in removing variance of Y_i's that is shared by $Y_{i,1}$ and $Y_{i,2}$ but cannot be explained by the treatment and covariates. If a unit has some hidden individual characteristics, those characteristics could influence both $Y_{i,1}$ and $Y_{i,2}$ in a similar fashion but may not be well captured by the observed covariates. To make this intuition precise, we present Algorithm 2, which makes uses of information from two experiments and tests for the existence of interference effect. We have also included an illustration of the algorithm in Figure 2.

Algorithm 2 has a few key differences from Algorithm 1. First, the choices of focal units are different. In Algorithm 1, the choice of the focal units cannot depend on the treatment assignments $W_{1:n}$, whereas in Algorithm 2, the focal units are randomly chosen from those whose treatment didn't change. This specific choice guarantees that the treatment of the ith unit will not influence
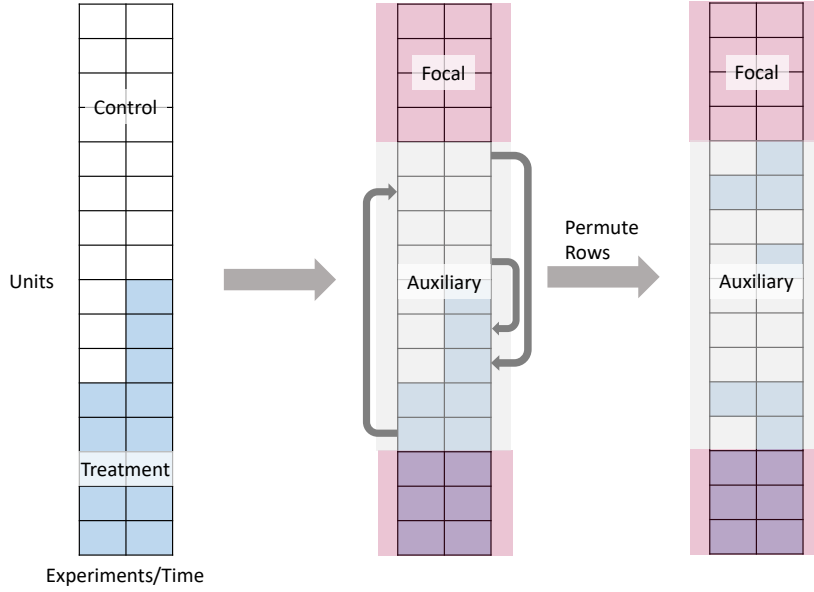
Figure 2: An illustration of Algorithm 2. After selecting the set of focal units and auxiliary units, we randomly permute rows of the treatment matrix and compute test statistics and p-values based on the permuted data.

the difference of $Y_{i,2}$ and $Y_{i,1}$ much. Second, as mentioned above, in computing the test statistics, Y^{diff} is used instead of Y itself. As explained above, this helps reduce variance. Third, instead of regenerating treatment, Algorithm 2 permutes the treatment of the auxiliary units. This change is necessary to guarantee the procedure's validity; the choice of focal units depends on the treatment vector, and thus naively regenerating treatments will not give a valid procedure anymore. This will be demonstrated in Section 4.

3.2 Testing with a time fixed effect model

In the previous section, we allow the existence of "arbitrary time effect". In particular, Hypothesis 1 allows the outcome $Y_{i,k}$ to depend on the treatments in other experiments, and does not restrict the relationship among outcomes in different experiments. This brings flexibility and generality, but it could reduce the power of the testing procedures. In this section, we make additional assumptions on the structure of time effect and propose a different testing procedure.

Assumption 1 (No temporal interference). $Y_{i,k}(w_{1:n,1:K}) = Y_{i,k}(\tilde{w}_{1:n,1:K})$ if $w_{1:n,k} = \tilde{w}_{1:n,k}$.

Assumption 1 states that the outcomes in experiment k depends only on treatments assigned in experiment k. In other words, the effect of treatment in one experiment will not carry over to the other experiments. Under Assumption 1, we can simplify the notation of potential outcomes: for any $w_{1:n} \in \{0,1\}^n$, we write $Y_{i,k}(w_{1:n})$ as the potential outcome and assume that the observed outcomes satisfy $Y_{i,k} = Y_{i,k}(W_{1:n,k})$. Note the difference from the previous notation. Previously, we wrote the potential outcomes $Y_{i,k}(w_{1:n,1:K})$ for any $w_{1:n,1:K} \in \{0,1\}^{n \times K}$. Here we focus on the potential outcomes $Y_{i,k}(w_{1:n})$ for any $w_{1:n} \in \{0,1\}^n$. Following this new notation, we make an additional assumption.

8

Assumption 2 (Time fixed effect). For any $w_{1:n} \in \{0,1\}^n$, $i \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, K\}$, $Y_{i,k}(w_{1:n}) = \alpha_i(w_{1:n}) + u_k + \epsilon_{i,k}(w_{1:n})$. The random variables $\epsilon_{i,1}(w_{1:n}), \ldots, \epsilon_{i,K}(w_{1:n})$ are zero mean, and are independently and identically distributed, independently of functions $\alpha_{1:n}$, variables $u_{1:K}$, treatments $W_{1:n,1:K}$, covariates $X_{1:n}$ and other errors $\epsilon_{j,l}$ for $j \neq i$.

Assumption 2 assumes a time fixed effect model. The term u_k captures the time effect: some special events may happen when the kth experiment is implemented, and Assumption 2 assumes that the effect of such events is shared by all units in the experiments. The term $\alpha_i(w)$ captures the individual effect, which could depend on the treatment of unit i as well as treatments of other units. Finally, the terms $\epsilon_{i,k}(w_{1:n})$'s are errors that are i.i.d. across experiments.

We also note that the commonly used *no temporal effect* assumption is a special case (stronger version) of Assumption 2. The no temporal effect assumption assumes that $Y_{i,k}(w_{1:n}) = \alpha_i(w_{1:n}) + \epsilon_{i,k}(w_{1:n})$, where the errors $\epsilon_{i,k}(w_{1:n})$'s are zero mean and i.i.d. across experiments. This corresponds to Assumption 2 with all time fixed effects $u_k = 0$. The no temporal effect assumption is particularly plausible when all the experiments are implemented within a short period of time, where the distribution of $Y_{i,k}(w_{1:n})$ is not expected to change by much.

Assumption 1 and Hypothesis 1 together state that the outcome $Y_{i,k}$ depend only on the treatment of unit i in experiment k. Therefore, under Assumption 1 and Hypothesis 1, we can further simplify the notation of potential outcomes: for any $w \in \{0,1\}$ we write $Y_{i,k}(w)$ as the potential outcome and assume that the observed outcomes satisfy $Y_{i,k} = Y_{i,k}(W_{i,k})$.[3] With this new notation, Assumption 2, together with Assumption 1 and Hypothesis 1, becomes a new hypothesis:

Hypothesis 1'. For any $w \in \{0,1\}$, $i \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, K\}$,

$$Y_{i,k}(w) = \alpha_i(w) + u_k + \epsilon_{i,k}(w), \tag{6}$$

such that the vectors $\epsilon_{1:n,1}(w), \ldots, \epsilon_{1:n,K}(w)$ are independently and identically distributed, independently of functions $\alpha_{1:n}$, vector $u_{1:K}$, treatments $W_{1:n,1:K}$, covariates $X_{1:n}$ and other errors $\epsilon_{j,l}(w)$ for $l \neq k$.

This corresponds to the two-way ANOVA in statistics literature [Yates, 1934, Fujikoshi, 1993] and the two-way fixed effect model in economics literature [Bertrand et al., 2004, Angrist and Pischke, 2009].

In the previous section, we conduct some permutation tests that permute the data "vertically", i.e., permute different units. Here with the additional assumptions, we can conduct permutation tests that permute the data "horizontally", i.e., permute different time points or experiments.

To motivate the permutation test, consider two units i and j. Assume that i has been in the treatment group the whole time, while j has been in the control group the whole time. Under Hypothesis 1', we have for the first experiment, $Y_{i,1} - Y_{j,1} = (\alpha_i(1) + u_1 + \epsilon_{i,1}(1)) - (\alpha_j(0) + u_1 + \epsilon_{j,1}(0)) = \alpha_i(1) - \alpha_j(0) + \epsilon_{i,1}(1) - \epsilon_{j,1}(0)$, and for the second experiment, $Y_{i,2} - Y_{j,2} = (\alpha_i(1) + u_2 + \epsilon_{i,2}(1)) - (\alpha_j(0) + u_1 + \epsilon_{j,1}(0)) = \alpha_i(1) - \alpha_j(0) + \epsilon_{i,2}(1) - \epsilon_{j,2}(0)$. Thus,

$$Y_{i,1} - Y_{j,1} = \alpha_i(1) - \alpha_j(0) + \epsilon_{i,1}(1) - \epsilon_{j,1}(0) \stackrel{d}{=} \alpha_i(1) - \alpha_j(0) + \epsilon_{i,2}(1) - \epsilon_{j,2}(0) = Y_{i,2} - Y_{j,2}. \tag{7}$$

---

[3]Note again the difference with the previous notation. Here we focus on the potential outcomes $Y_{i,k}(w)$ for any $w \in \{0,1\}$, while we consider $w_{1:n,1:K} \in \{0,1\}^{n \times K}$ for the most general case and $w_{1:n} \in \{0,1\}^n$ assuming Assumption 1.

Algorithm 3 Testing for interference effect (two experiments, time fixed effect model).

Input: Datasets $\mathcal{D}_1 = (W_{1:n,1}, X_{1:n}, Y_{1:n,1}, H_{1:n,1})$, $\mathcal{D}_2 = (W_{1:n,2}, X_{1:n}, Y_{1:n,2}, H_{1:n,2})$, matching algorithm m, test statistic T.

1. Let $\mathcal{I}_0 = \{i : W_{i,1} = W_{i,2} = 0\}$ and $\mathcal{I}_1 = \{i : W_{i,1} = W_{i,2} = 1\}$.
2. For each i in $\mathcal{I}_1$, match an index $j \in \mathcal{I}_0$ to i (with no repeat): let $m(i)$ be the matched index of i. Let $\mathcal{I}_m = \{m(i) : i \in \mathcal{I}_1\}$ be the set of matched indices.[4]
3. For each $k \in \{1, 2\}$, compute $Y_{\mathcal{I}_1,k}^{\text{diff}} = (Y_{i,k} - Y_{m(i),k})_{i \in \mathcal{I}_1}$, which is the vector of differences between the outcomes of the treated units and those of the matched units.
   Compute a test statistic $T^{(0)} = T(Y_{\mathcal{I}_1,1:2}^{\text{diff}}, X_{\mathcal{I}_m}, H_{\mathcal{I}_m,1:2}, X_{\mathcal{I}_1}, H_{\mathcal{I}_1,1:2})$.
4. **For** $b = 1, \ldots B$:
     For each $i \in \mathcal{I}_1$:
         Randomly permute outcomes across experiments: $\widetilde{Y}_{i,k}^{(b)} = Y_{i,\sigma_{i,b}(k)}$ and
         $\widetilde{Y}_{m(i),k}^{(b)} = Y_{m(i),\sigma_{i,b}(k)}$ for some permutation $\sigma_{i,b}$ of $\{1, 2\}$.
     End For
     Recompute $\widetilde{Y}_{\mathcal{I}_1,k}^{\text{diff},(b)} = (\widetilde{Y}_{i,k}^{(b)} - \widetilde{Y}_{m(i),k}^{(b)})_{i \in \mathcal{I}_1}$.
     Recompute the test statistic: $T^{(b)} = T(\widetilde{Y}_{\mathcal{I}_1,1:2}^{\text{diff}(b)}, X_{\mathcal{I}_m}, H_{\mathcal{I}_m,1:2}, X_{\mathcal{I}_1}, H_{\mathcal{I}_1,1:2})$.
   End For

Output: The p-value

$$p = \frac{1}{B+1}\left(1 + \sum_{b=1}^{B} \mathbb{1}\left\{T^{(0)} \leq T^{(b)}\right\}\right). \tag{9}$$

To put it simply, under Hypothesis 1', $Y_{i,1} - Y_{j,1}$ has the same distribution as $Y_{i,2} - Y_{j,2}$. However, when there is cross-unit interference, the two distributions could be different. Consider a simple model:

$$Y_{i,k} = W_{i,k}H_{i,k} + \epsilon_{i,k}, \tag{8}$$

where $H_{i,k}$ is the fraction of neighbors of unit i treated in experiment k, and $\epsilon_{i,k}$'s are some i.i.d. zero mean errors. Under this model, $Y_{i,1} - Y_{j,1} = H_{i,1} + \epsilon_{i,1} - \epsilon_{j,1}$ and $Y_{i,2} - Y_{j,2} = H_{i,2} + \epsilon_{i,2} - \epsilon_{j,2}$. When the number of neighbors of unit i is large, by law of large numbers, we have $H_{i,1} \approx \pi_1$ and $H_{i,2} \approx \pi_2$. We can then observe that $Y_{i,1} - Y_{j,1}$ and $Y_{i,2} - Y_{j,2}$ have different distributions; in particular, they have different means.

Given the above observation, we can conduct a permutation test permuting pairs of (i, j) across experiments. We outline the algorithm in Algorithm 3. We also provide an illustration of Algorithm 3 in Figure 3.

In Algorithm 3, we compare the value of a test statistic to the value of the statistic after permutation. One simple choice of test statistic is the difference-in-differences statistic:

$$T(Y_{\mathcal{I}_1,1:2}^{\text{diff}}, X_{\mathcal{I}_m}, H_{\mathcal{I}_m,1:2}, X_{\mathcal{I}_1}, H_{\mathcal{I}_1,1:2}) = \left|\text{mean}(Y_{\mathcal{I}_1,2}^{\text{diff}}) - \text{mean}(Y_{\mathcal{I}_1,1}^{\text{diff}})\right|, \tag{10}$$

where $\mathcal{I}_1$ and $\mathcal{I}_m$ are defined in the first step of Algorithm 3. We use the simple model (8) discussed above to explain why this choice of statistic is reasonable. Under model (8), the difference-in-differences statistic (without absolute value) will be

$$\text{mean}(Y_{\mathcal{I}_1,2}^{\text{diff}}) - \text{mean}(Y_{\mathcal{I}_1,1}^{\text{diff}}) \approx \text{mean}(H_{\mathcal{I}_1,2}) - \text{mean}(H_{\mathcal{I}_1,1}) \approx \pi_2 - \pi_1. \tag{11}$$

---

[4]Here we assume that $|\mathcal{I}_1| < |\mathcal{I}_0|$. If $|\mathcal{I}_1| \geq |\mathcal{I}_0|$, we start with $\mathcal{I}_1$ instead.
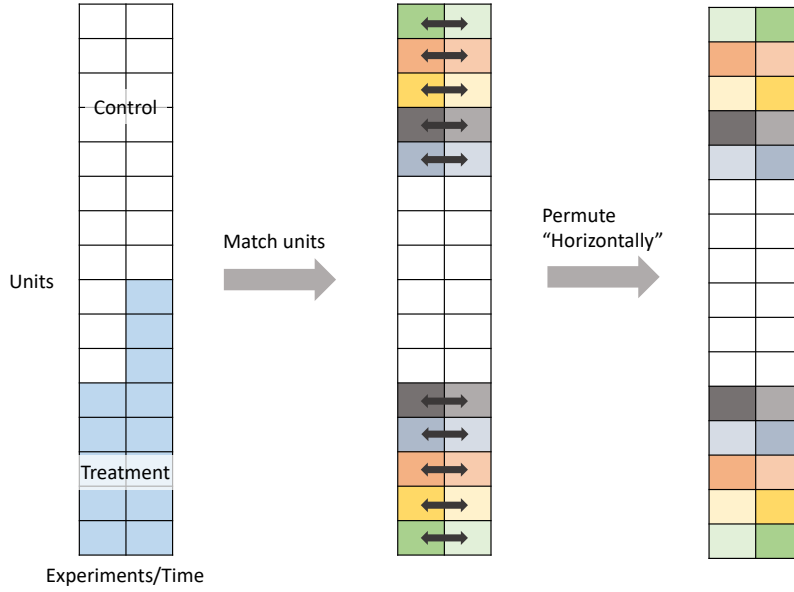
Figure 3: An illustration of Algorithm 3. Algorithm 3 permutes the outcomes horizontally (across experiments), whereas Algorithm 2 permutes the treatments vertically (across units).

However, after permutation, the difference-in-differences statistic (without absolute value) will be mean zero. Therefore, $T^{(0)}$ and $T^{(b)}$ will have different distributions and thus the p-value will be far from the Unif$[0, 1]$ distribution.

One advantage of this difference-in-differences test statistic is its simplicity. To compute this statistic, there is no need of constructing a candidate exposure or any interference graph, and thus the computation cost of the test statistic is very low. This test statistic is also very intuitive to understand. Recall the motivating example in Section 1.1: when the difference-in-means estimators are different, the difference-in-differences test statistic is large. With this test statistic, our algorithm formalizes the intuition of the motivating example in Section 1.1.

The difference-in-differences statistic is not the only one we can choose. Indeed, just as for Algorithms 1 and 2, we have full flexibility in choosing the test statistic. For example, we can add covariate adjustment into the test statistics: instead of taking the difference of $\mathrm{mean}(Y_{\mathcal{I}_1,2}^{\mathrm{diff}})$ and $\mathrm{mean}(Y_{\mathcal{I}_1,1}^{\mathrm{diff}})$, we can take the difference of the fitted intercepts after regressing Y_1^{diff} (and Y_2^{diff}) on $X_{\mathcal{I}_m}$ and $X_{\mathcal{I}_1}$. We can also bring the candidate exposure H into the picture. For example, we can similarly define $H_{\mathcal{I}_1,k}^{\mathrm{diff}} = \left(H_{i,k} - H_{m(i),k}\right)_{i \in \mathcal{I}_1}$ for $k \in \{1, 2\}$. Then one plausible test statistic (when $H_{i,k} \in \mathbb{R}$) is the following:

$$T(Y_{\mathcal{I}_1,1:2}^{\mathrm{diff}}, X_{\mathcal{I}_m}, H_{\mathcal{I}_m,1:2}, X_{\mathcal{I}_1}, H_{\mathcal{I}_1,1:2}) = \left| \mathrm{Corr}\left[Y_{\mathcal{I}_1,2}^{\mathrm{diff}} - Y_{\mathcal{I}_1,1}^{\mathrm{diff}}, H_{\mathcal{I}_1,2}^{\mathrm{diff}} - H_{\mathcal{I}_1,1}^{\mathrm{diff}} \right] \right|. \tag{12}$$

Finally, we want to comment on the matching algorithm m used in Algorithm 3. We would first like to stress that as long as the matching algorithm only looks at the covariates X, the test will be valid regardless of the quality of matching. In the most extreme case, we can simply conduct a random matching, and the test will remain valid. More ideally, we would hope each i is matched to an $m(i)$ such that X_i is close to $X_{m(i)}$. This matching step helps reduce variance due to the covariates and

thus increase the power of the test. In the causal inference literature, matching algorithms have been widely studied [Rubin, 1973, Stuart, 2010], and we recommend that experimenters choose from existing algorithms based on their needs and the computational resources available.

3.3 Usage of graphs of experimental units

In implementing the previously proposed algorithms, we often find it helpful to construct a graph of the n experimental units. Formally, let $G = (V, E)$, with vertex set $V = \{1, 2, \ldots, n\}$ and edge set $E = \{E_{ij}\}_{i,j=1}^{n}$. We will discuss a few different ways of using graphs to test and learn interference structure.

Interference graph. A graph can be constructed to model interference and to help compute candidate exposure. We call such a graph an *interference graph*. When experimental units are users, it is plausible to assume that a user's behavior is mostly influenced by friends in a social network. In this case, we can simply take the interference graph to be the social network, i.e., we set $E_{ij} = 1$ if user i and j are friends on the social network. With this graph, many candidate exposures can be computed easily: number of treated friends $H_{i,k}^{\mathrm{numFrds}} = \sum_{j:E_{ij}=1} W_{j,k}$, fraction of friends that are treated $H_{i,k}^{\mathrm{fracFrds}} = \sum_{j:E_{ij}=1} W_{j,k} / |\{j : E_{ij} = 1\}|$, number of treated two-hop friends $H_{i,k}^{\mathrm{num2Frds}} = \sum_{l:\exists j \text{ s.t.} E_{ij}E_{jl}=1} W_{j,k}$. The interference graph can be constructed differently in other settings. When experimental units are advertisers, there is no natural social network. However, we can construct a "competition network" based on the similarity of the covariates. For a similarity measure s and a threshold ϵ, we can define $E_{ij} = \mathbb{1}\{s(X_i, X_i) \geq \epsilon\}$. Such a graph reflects that an advertiser is mostly influenced by its competitors, especially those that are similar to it. Candidate exposures can then be computed based on this interference graph: number of treated competitors $H_{i,k}^{\mathrm{numCpt}} = \sum_{j:E_{ij}=1} W_{j,k}$, weighted average of competitors' treatments: $H_{i,k}^{\mathrm{wAvgCpt}} = \sum_{j:E_{ij}=1} s(X_i, X_i) W_{j,k}$.

The interference graph also helps experimenters to understand the nature of interference. Imagine we have two different interference graphs G_1 and G_2 and we apply the testing procedure separately using G_1 and G_2. If we observe a much smaller p-value for the procedure using G_1 than that we obtain using G_2, then we have some evidence suggesting that the interference in the form of G_1 is much stronger than in that of G_2. In particular, the units that are connected to unit i in G_1 might be the most influential in impacting the outcome of unit i. This kind of analysis, though not fully rigorous, can help experimenters to build better intuitions for modelling in subsequent analysis. For example, once the interference effect is statistically significant, experimenters may consider re-running experiments with a cluster randomized controlled trial. Understanding the structure of interference can be helpful in constructing better clusters.

Graph in matching. A graph can also be helpful in the matching step in Algorithm 3. In the causal inference literature, matched pairs are often constructed using a minimum cost flow algorithm on a bipartite graph with treated units on one side and control units on the other side [Rosenbaum, 1989, Hansen and Klopfer, 2006]. Here, the cost of flow from unit i to j can be defined as some dissimilarity metric between X_i and X_j. For example, the Mahalanobis distance is a common choice of such a dissimilarity metric [Rubin, 1980]. The bipartite graph may not always be a complete bipartite graph: sometimes a caliper can be applied to the graph resulting in the removal of edges. A caliper based on covariates limits with which a unit can be paired [Mahmood,

2018].[5] For example, researchers may only want advertisers to be matched/paired with advertisers who sell products of the same category; in such cases, there is an edge between i and j only if they sell products of the same category.

Interestingly, calipered graphs may correspond to the interference graph introduced in the above section, and thus we only need to construct the graph once and use it in both the step of computing candidate exposure and the step of matching. This is especially relevant in a market competition application: a company is expected to be mostly influenced by companies selling similar products, and thus we put edges in the interference graph; in the meantime, we would like to match companies selling similar products, and thus we put edges in the bipartite graph used in matching.

3.4 Aggregating p-values

One issue with the algorithms above proposed is that randomly splitting the data (Algorithms 1 and 2) or the random matching step (Algorithm 3) can inject randomness into the p-value. In order to derandomize the procedure, we can run the algorithms many times and aggregate the p-values. Since the p-values can be arbitrarily dependent on each other, we cannot use Fisher's method to aggregate the p-values, which requires independence [Fisher, 1925]. Some possible ways include, e.g., setting $p = 2 \sum p_i / n$ (See Vovk and Wang [2020] for more details).

In the previous section, we discuss the usage of an interference graph in constructing candidate exposure. In practice, experimenters may construct several interference graphs with different sparsity or structure. We can make use of information from different graphs and construct an "aggregated p-value". We can run the algorithms separately for each graph, and compute an "aggregated test statistic". For example, we can choose $T^{\text{aggre}} = \sum_m T(G_m)$, where G_m is the mth interference graph considered. Then we can compute an aggregated p-value in the following way:

$$p^{\text{aggre}} = \frac{1}{B+1} \left(1 + \sum_{m=1}^{B} \mathbb{1} \left\{ T^{\text{aggre}} \leq T^{\text{aggre}(b)} \right\} \right). \tag{13}$$

3.5 Extension to three or more experiments

More generally, experiments may be conducted more than two times. Formally, suppose that we run K experiments where treatments are randomly assigned according to (1) and (2). To test for interference, we can adopt a similar strategy as in Section 3.1. We outline the general algorithm in Algorithm 4. We note that Algorithm 2 is a special case of Algorithm 4. In practice, we recommend computing the test statistic using the difference of outcomes between experiments (as emphasized in Algorithm 2), since this helps remove common variance shared by outcomes in the experiments. One example of such statistic is the following.

$$T(W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, H_{\text{foc},1:K}) = \sum_{(k,l):k \neq l} \left| \text{Corr} \left[Y_{\text{foc},k} - Y_{\text{foc},l}, H_{\text{foc},l} - H_{\text{foc},l} \right] \right|. \tag{14}$$

If we assume a time fixed effect model as in Section 3.2, we can then extend Algorithm 3 to settings with more experiments. We outline the algorithm in Algorithm 5. Again, we note that Algorithm 3 is a special case of Algorithm 5. Algorithm 5 allows permutation over more experiments than

---

[5]In the observational study literature, calipers are often applied on the propensity score [Cochran and Rubin, 1973, Rosenbaum and Rubin, 1985]. Here we are in an experimental setting instead, where the propensity score is known, and it is the same for all units.

Algorithm 4 Testing for interference effect (multiple experiments).

Input: Datasets $\mathcal{D}_k = (W_{1:n,k}, X_{1:n}, Y_{1:n,k}, H_{1:n,k})$ for $k = 1, \ldots, K$, exposure function h, test statistic T.

1. Let $\mathcal{I}_{\mathrm{nc}} = \{i : W_{i,1} = \cdots = W_{i,K}\}$ be the set of units whose treatment didn't change over the experiments. Randomly sample a subset of $\mathcal{I}_{\mathrm{nc}}$ of size $n/2$. We call the subset $\mathcal{I}_{\mathrm{foc}}$. Let $\mathcal{I}_{\mathrm{aux}} = [n] \setminus \mathcal{I}_{\mathrm{foc}}$.
2. Compute a test statistic $T^{(0)} = T(W_{\mathrm{foc},1:K}, X_{\mathrm{foc}}, Y_{\mathrm{foc},1:K}, H_{\mathrm{foc},1:K})$ that captures the importance of H in predicting Y.
3. **For** $b = 1, \ldots B$:

    Randomly permute treatments for the auxiliary units of the data: $\widetilde{W}_{i,1:K}^{(b)} = W_{\sigma^{(b)}(i),1:K}$ for $i \in \mathcal{I}_{\mathrm{aux}}$, for some permutation $\sigma^{(b)}$ of $\mathcal{I}_{\mathrm{aux}}$.

    Recompute the candidate exposure for the focal units: $\widetilde{H}_{i,k}^{(b)} = h_i(W_{\mathrm{foc}\setminus\{i\},k}, \widetilde{W}_{\mathrm{aux},k}^{(b)})$, for $i \in \mathcal{I}_{\mathrm{foc}}$ and $k \in \{1, 2, \ldots, K\}$.

    Recompute the test statistic: $T^{(b)} = T(W_{\mathrm{foc},1:K}, X_{\mathrm{foc}}, Y_{\mathrm{foc},1:K}, \widetilde{H}_{\mathrm{foc},1:K}^{(b)})$.

    End For

Output: The p-value

$$p = \frac{1}{B+1}\left(1 + \sum_{b=1}^{B} \mathbb{1}\left\{T^{(0)} \leq T^{(b)}\right\}\right). \tag{15}$$

Algorithm 3 does. In particular, if unit i is treated in experiments $K_1, K_1 + 1, \ldots, K$, then the algorithm permutes outcome for unit i and its matched unit over experiments $K_1, K_1 + 1, \ldots, K$. Permuting over more experiments helps the test to leverage information from more experiments and thus increases power of the test. We have included an illustration of this algorithm in Figure 4.

4 Validity of the testing procedures

In this section, we establish validity of the above proposed algorithms. We make use of the following theorem in Hemerik and Goeman [2018a,b, Theorem 2].

Theorem 1 (Random permutations). *Let* $A_1, A_2, \ldots, A_n \in \mathcal{A}$ *be* n *random variables. Let* $\mathcal{S}_n$ *denote the set of all permutations on* $[n]$*. Assume that*

1. *$G \subset \mathcal{S}_n$ is a subgroup;*

2. *For any $\sigma \in G$, $A = (A_1, \ldots, A_n) \overset{d}{=} (A_{\sigma(1)}, \ldots, A_{\sigma(n)}) = A_\sigma$.*

If $\sigma_1, \ldots, \sigma_B$ are drawn independently uniformly from G, then for any test statistic T, the p-value

$$p = \frac{1}{B+1}\left(1 + \sum_{b=1}^{B} \mathbb{1}\left\{T(A) \leq T(A_\sigma)\right\}\right) \tag{17}$$

satisfies

$$\mathbb{P}\left[p \leq \alpha\right] \leq \alpha. \tag{18}$$

for any $\alpha \in (0, 1)$.

---
[6]Here we assume that $|\mathcal{I}_0| \geq n/2$.

Algorithm 5 Testing for interference effect (multiple experiments, time fixed effect model).

Input: Datasets $\mathcal{D}_k = (W_{1:n,k}, X_{1:n}, Y_{1:n,k}, H_{1:n,k})$ for $k = 1, \ldots, K$, matching algorithm m, test statistic T.

1. Let $\mathcal{I}_0 = \{i : W_{i,1} = \cdots = W_{i,k} = 0\}$ be the set of units that are in the control group in all experiments. Let $\mathcal{I}_1 = \{i : W_{i,K-1} = W_{i,K} = 1\}$ be the set of units that are in the treatment group in the last two experiments (i.e. units that are treated in at least two experiments).
2. For each i in $\mathcal{I}_1$, match an index $j \in \mathcal{I}_0$ to i (with no repeat): let $m(i)$ be the matched index of i. Let $\mathcal{I}_m = \{m(i) : i \in \mathcal{I}_1\}$ be the set of matched indices.[6]
3. For each $k \in \{1, \ldots, K\}$, compute $Y_{\mathcal{I}_1,k}^{\text{diff}} = \left(Y_{i,k} - Y_{m(i),k}\right)_{i \in \mathcal{I}_1}$, which is the vector of differences between the outcomes of the units in $\mathcal{I}_0$ and those of the matched units. Compute a test statistic $T^{(0)} = T(Y_{\mathcal{I}_1,1:K}^{\text{diff}}, X_{\mathcal{I}_m}, H_{\mathcal{I}_m,1:K}, X_{\mathcal{I}_1}, H_{\mathcal{I}_1,1:K})$.
4. **For** $b = 1, \ldots B$:
   > **For** each $i \in \mathcal{I}_1$:
   >> Let $S_i = \{k : W_{i,k} = 1\}$ be the set of experiments in which unit i is treated.
   >> Randomly permute outcomes across S_i: $\widetilde{Y}_{i,k}^{(b)} = Y_{i,\sigma_{i,b}(k)}$ and $\widetilde{Y}_{m(i),k}^{(b)} = Y_{m(i),\sigma_{i,b}(k)}$ for all $k \in S_i$, where $\sigma_{i,b}$ is a random permutation of S_i.
   >
   > **End For**
   > Recompute $\widetilde{Y}_{\mathcal{I}_1,k}^{\text{diff},(b)} = (\widetilde{Y}_{i,k}^{(b)} - \widetilde{Y}_{m(i),k}^{(b)})_{i \in \mathcal{I}_1}$.
   > Recompute the test statistic: $T^{(b)} = T(\widetilde{Y}_{\mathcal{I}_1,1:K}^{\text{diff}(b)}, X_{\mathcal{I}_m}, H_{\mathcal{I}_m,1:K}, X_{\mathcal{I}_1}, H_{\mathcal{I}_1,1:K})$.

   End For

Output: The p-value

$$p = \frac{1}{B+1}\left(1 + \sum_{b=1}^{B} \mathbb{1}\left\{T^{(0)} \leq T^{(b)}\right\}\right). \tag{16}$$
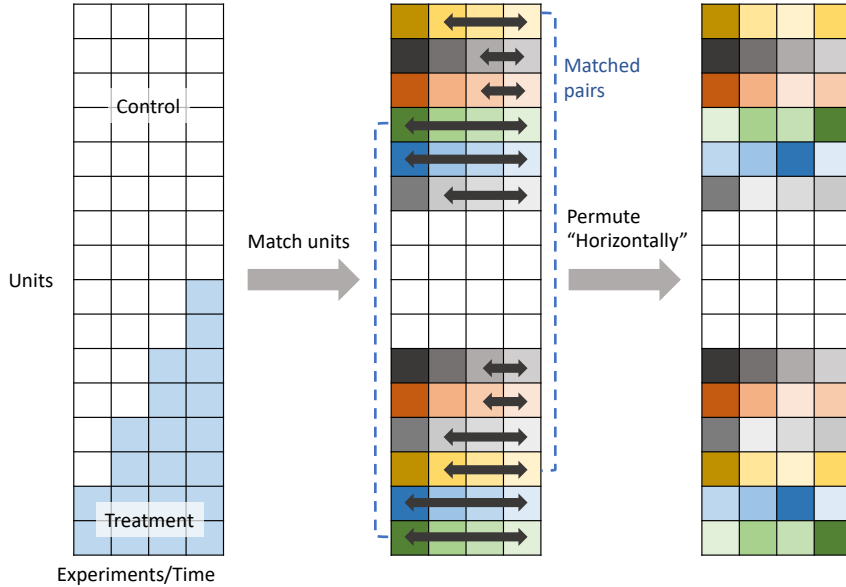

Figure 4: An illustration of Algorithm 5. Pairs of units are matched and the outcomes of paired units are permuted together across experiments. Test statistics and the p-value are then obtained based on the permuted data.

We start with establishing the validity of Algorithms 1, 2 and 4 under general assumptions.

Theorem 2. *Assume that the treatments are assigned according to rules defined in* (1) *and* (2). *Under Hypothesis 1, the p-values produced by Algorithms 1, 2 and 4 are valid in the following sense: for any $\alpha \in (0, 1)$,*

$$\mathbb{P}[p \leq \alpha] \leq \alpha. \tag{19}$$

Proof. Algorithm 1 has been shown to provide valid p-values in Athey et al. [2018]. Since Algorithm 2 is a special case of Algorithm 4, it suffices to prove that the p-values produced by Algorithm 4 are valid. We will be making use of Theorem 1 to show the result.

We start by noting that since $H_{\text{foc},1:K}$ is a function of $W_{\text{foc},1:K}$ and $W_{\text{aux},1:K}$, the test statistic $T(W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, H_{\text{foc},1:K})$ can be rewritten as

$$T(W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, H_{\text{foc},1:K}) = \check{T}(W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, W_{\text{aux},1:K}) \tag{20}$$

for some function $\check{T}$. Thus, we can also rewrite

$$T(W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, \widetilde{H}^{(b)}_{\text{foc},1:K}) = \check{T}(W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, \widetilde{W}^{(b)}_{\text{aux},1:K}). \tag{21}$$

By construction, $\widetilde{W}^{(b)}_{\text{aux},1:K}$ is a random permutation of the rows of $W_{\text{aux},1:K}$. Thus, we can take the permutation group G to be the set of all permutation on $\mathcal{I}_{\text{aux}}$. By Theorem 1, it suffices to establish that

$$W_{\sigma(\text{aux}),1:K} \mid W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, \mathcal{I}_{\text{foc}} \stackrel{d}{=} W_{\text{aux},1:K} \mid W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, \mathcal{I}_{\text{foc}}, \tag{22}$$

for any permutation $\sigma(\text{aux})$ on $\mathcal{I}_{\text{aux}}$. The above is equivalent to

$$\begin{aligned}
&\left(W_{\sigma(\text{aux}),1:K}, W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, \mathbb{1}\left\{\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}}\right\}\right) \\
&\qquad \stackrel{d}{=} \left(W_{\text{aux},1:K}, W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, \mathbb{1}\left\{\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}}\right\}\right),
\end{aligned} \tag{23}$$

for any fixed subset $\mathcal{I}_{\text{fix}} \subset [n]$ of size $n/2$. Let $\mathcal{I}_{\text{fix}^c} = [n] \setminus \mathcal{I}_{\text{foc}}$. Then, under the null hypothesis 1,

$$\begin{aligned}
&p\left(W_{\text{aux},1:K}, W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, \mathbb{1}\left\{\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}}\right\}\right) \\
&= p\left(W_{\text{fix}^c,1:K}, W_{\text{fix},1:K}, X_{\text{fix}}, Y_{\text{fix},1:K}, \mathbb{1}\left\{\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}}\right\}\right) \\
&= p(W_{\text{fix}^c,1:K})p(W_{\text{fix},1:K}, X_{\text{fix}}, Y_{\text{fix},1:K})\mathbb{P}\left[\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}} \mid W_{\text{fix}^c,1:K}, W_{\text{fix},1:K}\right],
\end{aligned} \tag{24}$$

where the last line follows from the no cross-unit interference hypothesis and the fact that treatments are sampled independently across units. Note also that permuting $\mathcal{I}_{\text{fix}^c}$ will not change the selection probability of the focal units, i.e., $\mathbb{P}\left[\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}} \mid W_{\text{fix}^c}, W_{\text{fix}}\right] = \mathbb{P}\left[\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}} \mid W_{\sigma(\text{fix}^c)}, W_{\text{fix}}\right]$, and thus

$$\begin{aligned}
&p(W_{\text{fix}^c,1:K})p(W_{\text{fix},1:K}, X_{\text{fix}}, Y_{\text{fix},1:K})\mathbb{P}\left[\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}} \mid W_{\text{fix}^c,1:K}, W_{\text{fix},1:K}\right] \\
&= p(W_{\sigma(\text{fix}^c),1:K})p(W_{\text{fix},1:K}, X_{\text{fix}}, Y_{\text{fix},1:K})\mathbb{P}\left[\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}} \mid W_{\sigma(\text{fix}^c),1:K}, W_{\text{fix},1:K}\right] \\
&= p\left(W_{\sigma(\text{fix}^c),1:K}, W_{\text{fix},1:K}, X_{\text{fix}}, Y_{\text{fix},1:K}, \mathbb{1}\left\{\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}}\right\}\right) \\
&= p\left(W_{\sigma(\text{aux}),1:K}, W_{\text{foc},1:K}, X_{\text{foc}}, Y_{\text{foc},1:K}, \mathbb{1}\left\{\mathcal{I}_{\text{foc}} = \mathcal{I}_{\text{fix}}\right\}\right),
\end{aligned} \tag{25}$$

and thus proving (23). $\qquad\square$

Theorem 3 (Time fixed effect model)**.** *Assume that the treatments are assigned according to rules defined in* (1) *and* (2)*. Under Assumptions* 1*-* 2 *and Hypothesis* 1*, the p-values produced by Algorithms* 3 *and* 5 *are valid in the following sense: for any $\alpha \in (0,1)$,*

$$\mathbb{P}\left[p \leq \alpha\right] \leq \alpha. \tag{26}$$

Proof. Algorithm 3 is a special case of Algorithm 5, and thus we will only work with Algorithm 5 here. We will again make use of Theorem 1 to show the result.

By construction, the elements in $\widetilde{Y}_{\mathcal{I}_1,1:K}^{\mathrm{diff},(b)}$ are a random permutation of the elements in $Y_{\mathcal{I}_1,1:K}^{\mathrm{diff}}$. The allowed permutations in Algorithm 5 clearly form a group. Specifically, the allowed permutations are defined by $\sigma = (\sigma_i)_{i \in \mathcal{I}_1}$, where each σ_i is a permutation of $S_i = \{k : W_{i,k} = 1\}$, and $\sigma(Y_{i,k}^{\mathrm{diff}}) = Y_{i,\sigma_i(k)}^{\mathrm{diff}}$. Following this notation, by Theorem 1, it suffices to show that for any allowed permutation σ,

$$\sigma(Y_{\mathcal{I}_1,1:K}^{\mathrm{diff}}) \mid W_{1:n,1:K}, X_{1:n}, \mathcal{I}_m \overset{d}{=} Y_{\mathcal{I}_1,1:K}^{\mathrm{diff}} \mid W_{1:n,1:K}, X_{1:n}, \mathcal{I}_m. \tag{27}$$

Under Assumptions 1 - 2 and Hypothesis 1, following (6), we can write $Y_{i,k}(w) = \alpha_i(w) + u_k + \epsilon_{i,k}(w)$. Therefore, for any $i \in \mathcal{I}_1$ and $k \in S_i$, we have that $Y_{i,k} = Y_{i,k}(1) = \alpha_i(1) + u_k + \epsilon_{i,k}(1)$. At the same time, for the matched unit of i, we have $W_{m(i),k} = 0$, and thus $Y_{m(i),k} = Y_{m(i),k}(0) = \alpha_{m(i)}(0) + u_k + \epsilon_{m(i),k}(0)$. The difference of the two satisfies

$$
\begin{aligned}
Y_{i,k}^{\mathrm{diff}} = Y_{i,k} - Y_{m(i),k} &= \alpha_i(1) + u_k + \epsilon_{i,k}(1) - \left(\alpha_{m(i)}(0) + u_k + \epsilon_{m(i),k}(0)\right) \\
&= \alpha_i(1) + \epsilon_{i,k}(1) - \alpha_{m(i)}(0) + \epsilon_{m(i),k}(0).
\end{aligned}
\tag{28}
$$

Under Assumption 2, we have that

$$
\begin{aligned}
&\left(\alpha_i(1) + \epsilon_{i,k}(1) - \alpha_{m(i)}(0) + \epsilon_{m(i),k}(0)\right) \mid W_{1:n,1:K}, X_{1:n}, \mathcal{I}_m, \alpha_{1:n} \\
&\overset{d}{=} \left(\alpha_i(1) + \epsilon_{i,\sigma_i(k)}(1) - \alpha_{m(i)}(0) + \epsilon_{m(i),\sigma_i(k)}(0)\right) \mid W_{1:n,1:K}, X_{1:n}, \mathcal{I}_m, \alpha_{1:n}
\end{aligned}
\tag{29}
$$

for any permutation σ_i of S_i, because the errors $\epsilon_{i,k}$ and $\epsilon_{i,\sigma_i(k)}$ are i.i.d conditioning on $W_{1:n,1:K}, X_{1:n}$ and $\alpha_{1:n}$ (and same for $\epsilon_{m(i),k}$ and $\epsilon_{m(i),\sigma_i(k)}$). In addition, since all the errors $\epsilon_{i,k}$'s are independent conditioning on $W_{1:n,1:K}, X_{1:n}$ and $\alpha_{1:n}$, we have that

$$
\begin{aligned}
&\left(\alpha_i(1) + \epsilon_{i,k}(1) - \alpha_{m(i)}(0) + \epsilon_{m(i),k}(0)\right)_{i \in \mathcal{I}_1} \mid W_{1:n,1:K}, X_{1:n}, \mathcal{I}_m, \alpha_{1:n} \\
&\overset{d}{=} \left(\alpha_i(1) + \epsilon_{i,\sigma_i(k)}(1) - \alpha_{m(i)}(0) + \epsilon_{m(i),\sigma_i(k)}(0)\right)_{i \in \mathcal{I}_1} \mid W_{1:n,1:K}, X_{1:n}, \mathcal{I}_m, \alpha_{1:n}.
\end{aligned}
\tag{30}
$$

Rewriting the above, we get

$$Y_{\mathcal{I}_1,1:K}^{\mathrm{diff}}, \alpha_{1:n} \mid W_{1:n,1:K}, X_{1:n}, \mathcal{I}_m \overset{d}{=} \sigma(Y_{\mathcal{I}_1,1:K}^{\mathrm{diff}}) \mid W_{1:n,1:K}, X_{1:n}, \mathcal{I}_m, \alpha_{1:n}, \tag{31}$$

which further implies (27) and hence gives the desired result.

$\square$

5 Simulations

In this section, we focus on a form of network interference. Specifically, we use a real-life social network to describe social interactions among units. We generate outcomes with some magnitude of network interference and evaluate our methods based on these generated outcomes. Our simulations

can be viewed as semi-synthetic experiments—we use a real-life network, but we generate outcomes according to some model.

We consider the Swarthmore network in the Facebook 100 dataset [Traud et al., 2012]. All networks in this dataset are complete online friendship networks for one hundred colleges and universities collected from a single-day snapshot of Facebook in September 2005. Here we focus on the Swarthmore college network in our simulation. To make the social network connected, we extract the largest connected component of the Swarthmore network. To summarize, the network we use is of size 1657 with 61049 edges. The diameter of the network is 6 and the average pairwise distance is 2.32.

Throughout this section, we assume that we have access to the data of three randomized experiments. We take treatment probabilities $\pi_1 = 10\%$, $\pi_2 = 25\%$ and $\pi_3 = 50\%$. In the following simulation studies, we consider level of significance $\alpha = 0.05$. Every dot on each plot is an average over 500 replications. We take $B = 200$.

5.1 Under general assumptions

We compare the power of the tests given in Algorithms 1, 2 and 4. We run Algorithm 4 using all three experiments, run Algorithm 2 using the second and the third experiments, and run Algorithm 1 using the third experiment, i.e., we always use experiments with the largest treatment probabilities. We discuss the choice of test statistics in Appendix A. In Figure 5a, we assume a linear model of the outcome Y; in Figure 5b, we assume a nonlinear model. The details of the generating model can also be found in Appendix A.

In Figures 5a and 5b, we plot the power of the testing algorithms 1, 2 and 4 at different levels of interference effects (signal strengths). In the figures, the fraction of common variance controls the correlation of the individual outcomes across experiments.

We observe from Figures 5a and 5b that utilizing more experiments helps our algorithms become more powerful, especially when the fraction of common variance is high. As discussed in Section 1.2, our work is the first to consider testing interference with multiple randomized experiments. Therefore, we can treat the algorithm utilizing one experiment as the *baseline method* that represents the state-of-the-art. Our algorithms appear to have a clear advantage over the baseline in terms of the power.
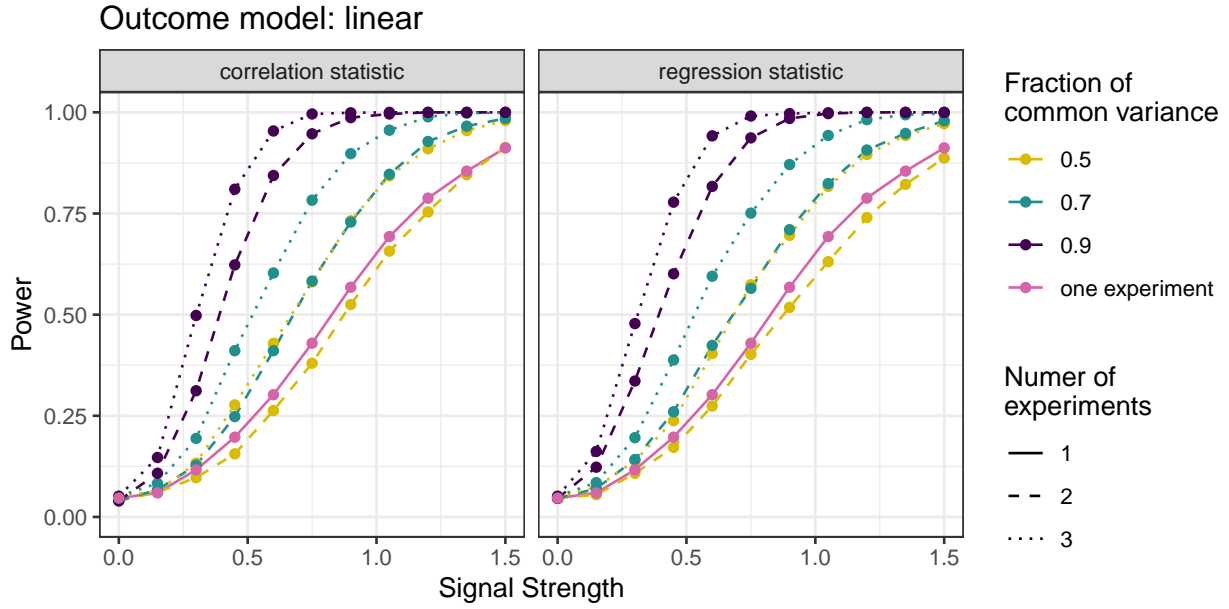
We also find that the regression statistic performs better than the correlation statistic, because the regression step helps reduce variance caused by the observed covariates.

5.2 Time fixed effect model

We compare the power of the tests given in Algorithms 4 and 5. We run both algorithms using all three experiments. We use a regression test statistic in both algorithms. We discuss the choice of test statistics and matching algorithms in Appendix A. In Figure 6a, we assume a linear model of the outcome Y, whereas in Figure 6b, we assume a nonlinear model. The details of the generating model can also be found in Appendix A.

In Figures 6a and 6b, we plot the power of the testing algorithms 4 and 5 at different levels of interference effects (signal strengths). Algorithm 5 (testing with a time fixed effect model) appears more powerful than Algorithm 4 (testing under general assumptions). To understand this phenomenon, we recall that Algorithm 4 permutes data across experiments, whereas Algorithm 5

(a) Outcome Y follows a linear model.

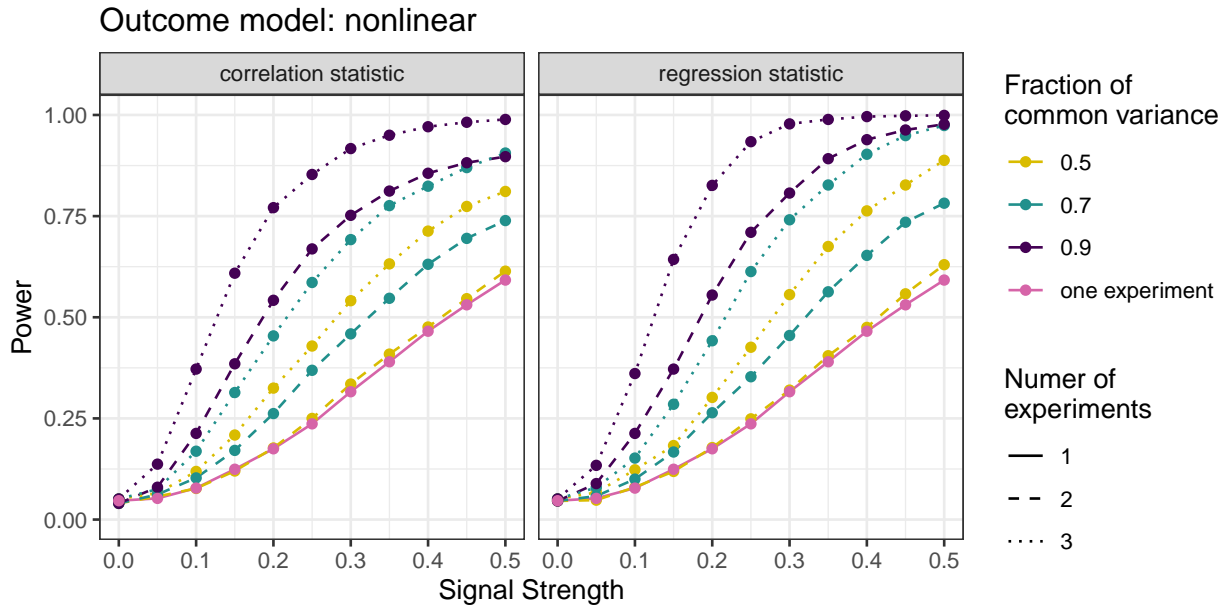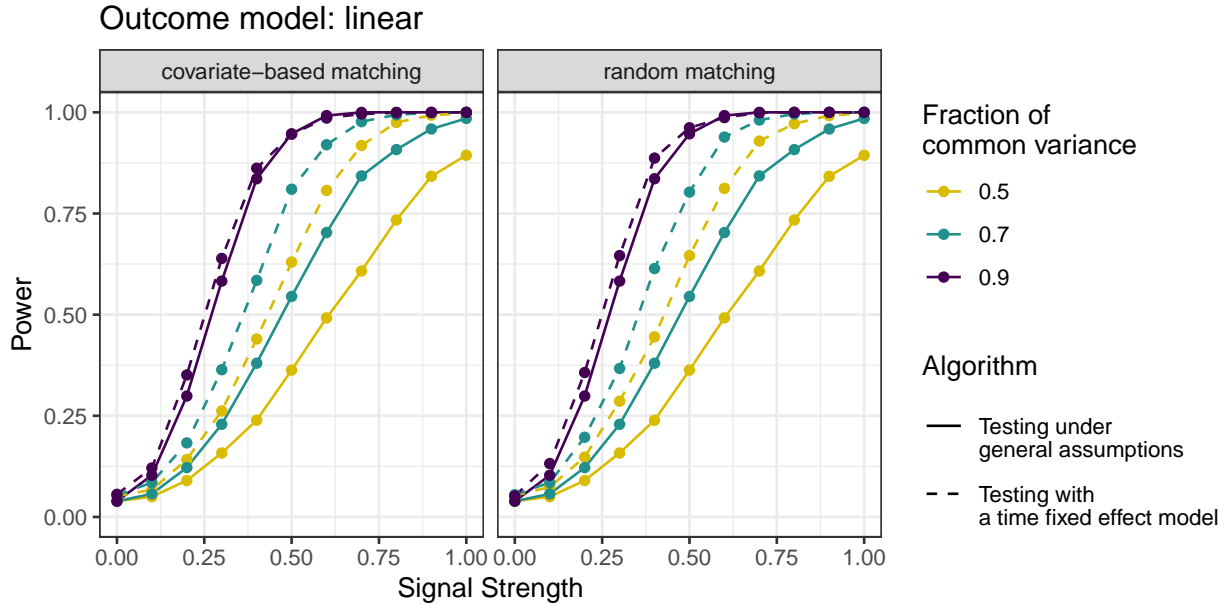

(b) Outcome Y follows a nonlinear model.

Figure 5: Power of Algorithms 1, 2 and 4.

(a) Outcome Y follows a linear model.
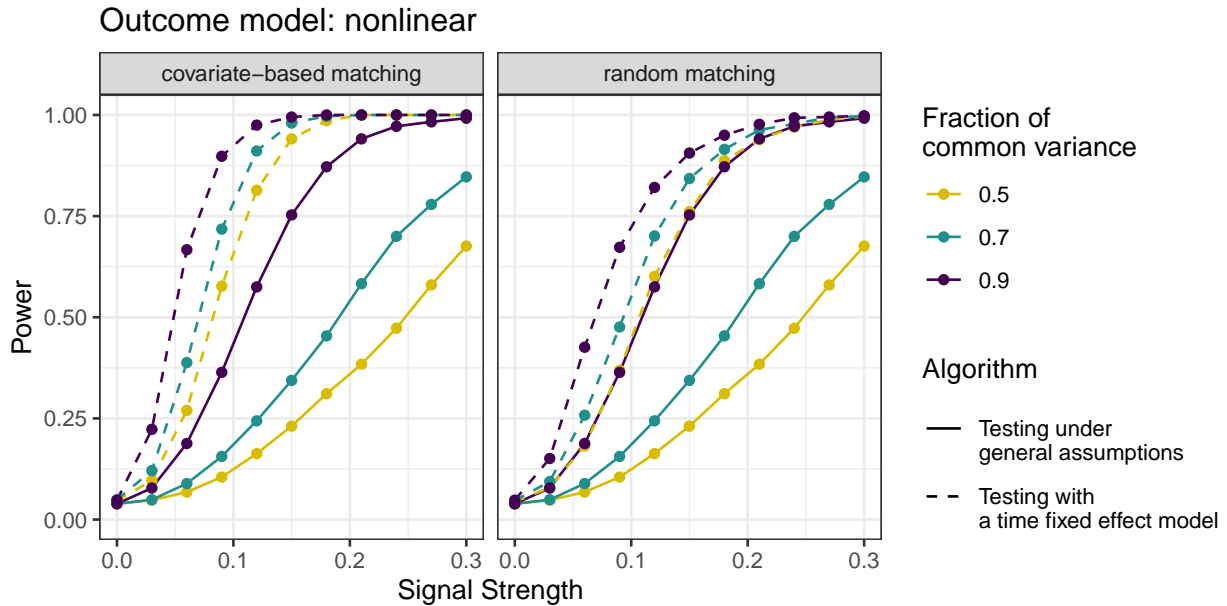

(b) Outcome Y follows a nonlinear model.

Figure 6: Power of Algorithms 4 and 5.

permutes data across units. Due to the nature of A/B tests, there is more variability in treatment allocation across experiments than across units. For example, assume that all units have around n_{ngb} neighbors in the social network. Looking at the fraction of neighbors in the treatment group, we find that the variation of this quantity across units is of scale $1/\sqrt{n_{\mathrm{ngb}}}$, whereas the variation of this quantity across experiments is of constant scale. By permuting over data points that are more different, Algorithm 5 gains extra power.

Recall that there is a matching step in Algorithm 5. We find from Figure 6a and 6b that covariate-based matching outperforms random matching, especially under a nonlinear outcome model. In a linear model, the regression step has already removed almost all of the variance caused by observed covariates. In a nonlinear model, nevertheless, the regression step cannot fully remove all variance and the matching step can help further reduce variance.

6 Applications

In this section, we illustrate how the proposed procedure has been successfully implemented at LinkedIn as an add-on to their experimentation toolkit. Like other firms in the technology sector such as Google and Meta, LinkedIn makes business decisions in a data-driven manner and has a culture to "test everything". To support the needs to run concurrent A/B tests at scale, LinkedIn built an in-house experimentation platform, called T-REX (Targeting, Ramping, and Experimentation), which provides end-to-end experimentation supports [Xu et al., 2015, Ivaniuk, 2020]. Regardless of the application, T-REX implements simple Bernoulli randomization and relies on t-test for readout without taking into account potential interactions among experimental units.

This becomes a major limitation for experimentation in a marketplace environment, including the ads marketplace, where units on either side of the marketplace (advertisers and ad viewers) can interfere with each other [Basse et al., 2016, Pouget-Abadie et al., 2019a, Liu et al., 2021, Johari et al., 2022]. For example, ad campaigns that share the targeting audiences interfere with each other by competing in auctions for ad slots; different ad viewers with similar attributes are connected through the finite budget of certain ad campaigns. To remove bias in experiments caused by interference, LinkedIn has implemented the Budget-split platform on top of T-REX for experimentation in their ads marketplace [Liu et al., 2021].

However, since Budget-split uses two halves of the marketplace to simulate the counterfactuals under different treatment variants, it does not support the classic factorial design. Under the current implementation, the platform only runs one experiment at a time, which is much smaller than the total number of experiments they need to run. This limitation in Budget-split capacity severely delays innovation: teams need to wait for weeks for a Budget-split slot in order to get an accurate measurement of their feature ramp before product launch. Nevertheless, not all ramps suffer from unit interaction, even in the ads marketplace setting. Running Budget-split experiments with negligible interference incurs a huge opportunity cost. Ideally, the Budget-split platform wants to prioritize tests that are impacted the most by the interference effects.

At LinkedIn, all feature launches start with small percentage ramps for risk mitigation and gradually increase the treatment percentage (i.e., 1%, 5%, 10%, 25%) before reaching the iteration for treatment effect measurement (50%) [Xu et al., 2018, Mao and Bojinov, 2021]. Specifically, Budget-split amounts to a 50% ramp on the viewers' side. This increasing allocation scheme provides us information to detect potential interference. With the algorithms proposed in this paper, we implemented a screening step for each feature after the 25% iteration. The experiments are
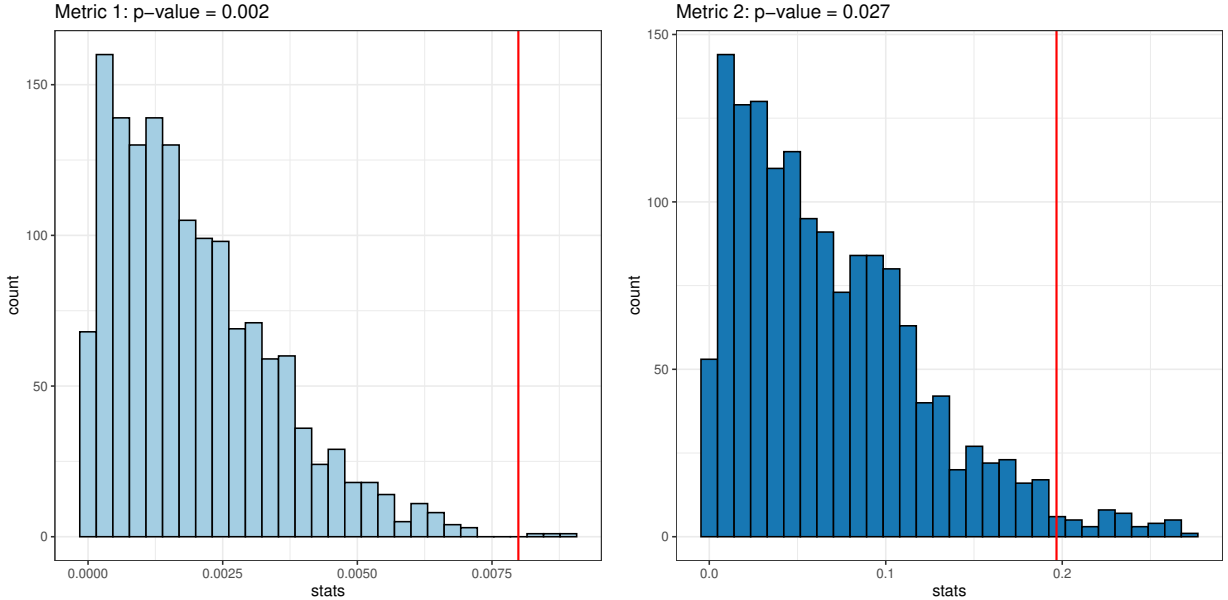
Figure 7: Example experiment: Test statistics and p-values from the permutation test. Results on two metrics are shown.

then ranked by the p-value in the interference test to determine their priority on the Budget-split platform.

It is important to note that the screening module was designed as an add-on to the system without touching LinkedIn's existing experimentation solution such as T-REX. By default, the interference detector only requires experimentation data in two previous iterations and runs Algorithm 3. Users have the option to provide additional network information that characterizes the potential interference mechanism among units and run other algorithms in this paper. Because of this standalone nature, a similar interference detector can be readily added to any existing experimentation platforms to trigger alerts when interference might cause a problem.

As an illustration, we consider an online controlled experiment implemented by LinkedIn. The treatment in this experiment corresponds to a new feature that improves the quality of LinkedIn members' attribute for ads targeting. We run a series of experiments with increasing allocation with the members as the randomization units. Interference effect is expected in these experiments: when the allocation percentage is small, only a small set of members have the updated attributes, making them easier to be targeted by ad campaigns. Thus, when comparing metrics such as total ad impressions, these members tend to have larger average results than members in the control group. When the treatment allocation increases, more members get the improved attributes. Since the total ad budget does not increase much, the average difference between treatment and control units becomes smaller. Figure 1 shows the average differences between treatment and control units in the experiment series. Figure 7 shows the output from the interference detector after running Algorithm 3 based on the 10% and 25% iterations with respect to two different metrics. The p-values of the permutation test confirm the strong interference effects in these experiments.

7 Discussion

Missingness. In this paper, we make the assumption that the dataset is complete. A natural future direction of work is to extend the current methods to scenarios with missing data. It is not hard to show that if the data is missing completely at random (MCAR), then the proposed testing procedures are still valid. When MCAR is unrealistic, it will be interesting to study whether our methods can still be applied under certain conditions. In practice, experimenters need to carefully examine the possible causes and consequences of missingness and make decisions correspondingly.

Selective inference. We propose to use our testing procedure as a screening step for A/B testing: if the test suggests that no interference exists, then the experimenter can proceed with classical causal inference analysis. Strictly speaking, the data is used twice here—in the screening step and in the follow-up analysis. It would be of interest to understand the impact of the screening step on the follow-up analysis, and to develop valid statistical inference methods conditioning on the result of the screening step.

Sequential Testing. Another question left open by this paper is whether the proposed methods can be extended to the sequential testing setting. Our current procedure fixes the number of experiments a priori and constructs a single p-value from the permutation test. In real life, the treatment probability increases gradually, and it would be of practical interest to end the experiment early as soon as we detect any interference. In that scenario, we need to take into account the randomness in stopping time and construct always valid p-values [Johari et al., 2017].

Acknowledgements

References

Joshua D Angrist and Jörn-Steffen Pischke. *Mostly harmless econometrics: An empiricist's companion.* Princeton university press, 2009.

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 and Cyrus Samii. 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.

Patrick Bajari, Brian Burdick, Guido W Imbens, Lorenzo Masoero, James McQueen, Thomas Richardson, and Ido M Rosen. Multiple randomization designs. *arXiv preprint arXiv:2112.13495*, 2021.

Eytan Bakshy, Dean Eckles, and Michael S Bernstein. Designing and deploying online field experiments. In *Proceedings of the 23rd international conference on World wide web*, pages 283–292, 2014.

G W Basse, A Feller, and P Toulis. Randomization tests of causal effects under interference. *Biometrika*, 106(2):487–494, 02 2019.

Guillaume Basse and Avi Feller. Analyzing two-stage experiments in the presence of interference. *Journal of the American Statistical Association*, 113(521):41–55, 2018. doi: 10.1080/01621459.2017.1323641.

Guillaume W. Basse and Edoardo M. Airoldi. Limitations of design-based causal inference and a/b testing under arbitrary and network interference. *Sociological Methodology*, 48(1):136–151, 2018. doi: 10.1177/0081175018782569.

Guillaume W Basse, Hossein Azari Soufiani, and Diane Lambert. Randomization and the pernicious effects of limited budgets on auction experiments. In *Artificial Intelligence and Statistics*, pages 1412–1420. PMLR, 2016.

Marianne Bertrand, Esther Duflo, and Sendhil Mullainathan. How much should we trust differences-in-differences estimates? *The Quarterly journal of economics*, 119(1):249–275, 2004.

Rohit Bhattacharya, Daniel Malinsky, and Ilya Shpitser. Causal inference under interference and network uncertainty. In *Uncertainty in Artificial Intelligence*, pages 1028–1038. PMLR, 2020.

Iavor Bojinov, Ashesh Rambachan, and Neil Shephard. Panel experiments and dynamic causal effects: A finite population perspective. *Quantitative Economics*, 12(4):1171–1196, 2021.

Jake Bowers, Mark M. Fredrickson, and Costas Panagopoulos. Reasoning about interference between units: A general framework. *Political Analysis*, 21(1):97–124, 2013. doi: 10.1093/pan/mps038.

William G Cochran and Donald B Rubin. Controlling bias in observational studies: A review. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 417–446, 1973.

Mayleen Cortez, Matthew Eichhorn, and Christina Lee Yu. Graph agnostic estimators with staggered rollout designs under network interference. *arXiv preprint arXiv:2205.14552*, 2022.

Dean Eckles, Brian Karrer, and Johan Ugander. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1):20150021, 2017.

Ronald Aylmer Fisher. *Statistical Methods for Research Workers*. Number 3. Oliver and Boyd, 1925.

Andrey Fradkin. A simulation approach to designing digital matching platforms. *Boston University Questrom School of Business Research Paper Forthcoming*, 2019.

Yasunori Fujikoshi. Two-way anova models with unbalanced data. *Discrete Mathematics*, 116(1-3):315–334, 1993.

Kevin Wu Han, Iavor Bojinov, and Guillaume Basse. Population interference in panel experiments, 2021. URL https://arxiv.org/abs/2103.00553.

Ben B Hansen and Stephanie Olsen Klopfer. Optimal full matching and related designs via network flows. *Journal of computational and Graphical Statistics*, 15(3):609–627, 2006.

Jesse Hemerik and Jelle Goeman. Exact testing with random permutations. *Test*, 27(4):811–825, 2018a.

Jesse Hemerik and Jelle J Goeman. False discovery proportion estimation by permutations: confidence for significance analysis of microarrays. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1):137–155, 2018b.

David Holtz, Ruben Lobel, Inessa Liskovich, and Sinan Aral. Reducing interference bias in online marketplace pricing experiments. *arXiv preprint arXiv:2004.12489*, 2020.

Yuchen Hu, Shuangning Li, and Stefan Wager. Average direct and indirect causal effects under interference. *Biometrika*, 02 2022. ISSN 1464-3510. doi: 10.1093/biomet/asac008. URL https://doi.org/10.1093/biomet/asac008. asac008.

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

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, 2015. doi: 10.1017/CBO9781139025751.

Alexander Ivaniuk. Our evolution towards t-rex: The prehistory of experimentation infrastructure at linkedin. *LinkedIn Engineering Blog*, 2020.

Ramesh Johari, Pete Koomen, Leonid Pekelis, and David Walsh. Peeking at A/B tests: Why it matters, and what to do about it. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1517–1525, New York, NY, USA, 2017. Association for Computing Machinery.

Ramesh Johari, Hannah Li, Inessa Liskovich, and Gabriel Y Weintraub. Experimental design in two-sided platforms: An analysis of bias. *Management Science*, 2022.

Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 1168–1176, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450321747. doi: 10.1145/2487575.2488217. URL https://doi.org/10.1145/2487575.2488217.

Ron Kohavi, Diane Tang, and Ya Xu. *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing*. Cambridge University Press, 2020. doi: 10.1017/9781108653985.

Michael P. Leung. Treatment and spillover effects under network interference. *The Review of Economics and Statistics*, 102(2):368–380, 05 2020. ISSN 0034-6535. doi: 10.1162/rest_a_00818.

Hannah Li, Geng Zhao, Ramesh Johari, and Gabriel Y Weintraub. Interference, bias, and variance in two-sided marketplace experimentation: Guidance for platforms. In *Proceedings of the ACM Web Conference 2022*, pages 182–192, 2022.

Shuangning Li and Stefan Wager. Random graph asymptotics for treatment effect estimation under network interference. *The Annals of Statistics*, 50(4):2334 – 2358, 2022. doi: 10.1214/22-AOS2191. URL https://doi.org/10.1214/22-AOS2191.

Min Liu, Jialiang Mao, and Kang Kang. Trustworthy and powerful online marketplace experimentation with budget-split design. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3319–3329, 2021.

Sharif Mahmood. The performance of largest caliper matching: A monte carlo simulation approach. *arXiv preprint arXiv:1806.02149*, 2018.

Jialiang Mao and Iavor Bojinov. Quantifying the value of iterative experimentation. *arXiv preprint arXiv:2111.02334*, 2021.

Jean Pouget-Abadie, Kevin Aydin, Warren Schudy, Kay Brodersen, and Vahab Mirrokni. Variance reduction in bipartite experiments through correlation clustering. *Advances in Neural Information Processing Systems*, 32, 2019a.

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

David Puelz, Guillaume Basse, Avi Feller, and Panos Toulis. A graph-theoretic approach to randomization tests of causal effects under general interference. *Journal of the Royal Statistical Society Series B*, 84(1):174–204, February 2022.

Paul R Rosenbaum. Optimal matching for observational studies. *Journal of the American Statistical Association*, 84(408):1024–1032, 1989.

Paul R Rosenbaum and Donald B Rubin. Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician*, 39(1):33–38, 1985.

Donald B Rubin. Matching to remove bias in observational studies. *Biometrics*, pages 159–183, 1973.

Donald B Rubin. Bias reduction using mahalanobis-metric matching. *Biometrics*, pages 293–298, 1980.

Martin Saveski, Jean Pouget-Abadie, Guillaume Saint-Jacques, Weitao Duan, Souvik Ghosh, Ya Xu, and Edoardo M Airoldi. Detecting network effects: Randomizing over randomized experiments. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1027–1035, 2017.

Fredrik Sävje, Peter M. Aronow, and Michael G. Hudgens. Average treatment effects in the presence of unknown interference. *The Annals of Statistics*, 49(2):673 – 701, 2021. doi: 10.1214/20-AOS1973. URL https://doi.org/10.1214/20-AOS1973.

Jasjeet S Sekhon. Multivariate and propensity score matching software with automated balance optimization: the matching package for r. *Journal of Statistical Software, Forthcoming*, 2008.

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

Elizabeth A Stuart. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1, 2010.

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.

Fredrik Sävje. Causal inference with misspecified exposure mappings, 2021. URL https://arxiv.org/abs/2103.06471.

Diane Tang, Ashish Agarwal, Deirdre O'Brien, and Mike Meyer. Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 17–26, 2010.

Eric J Tchetgen Tchetgen and Tyler J VanderWeele. On causal inference in the presence of interference. *Statistical Methods in Medical Research*, 21(1):55–75, 2012.

Panos Toulis and Edward Kao. Estimation of causal peer influence effects. In *International conference on machine learning*, pages 1489–1497. PMLR, 2013.

Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2011.12.021. URL https://www.sciencedirect.com/science/article/pii/S0378437111009186.

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*, KDD '13, page 329–337, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450321747.

Davide Viviano. Experimental design under network interference. *arXiv preprint arXiv:2003.08421*, 2020.

Vladimir Vovk and Ruodu Wang. Combining p-values via averaging. *Biometrika*, 107(4):791–808, 2020.

Stefan Wager and Kuang Xu. Experimenting in equilibrium. *Management Science*, 67(11):6694–6715, 2021.

Ya Xu, Nanyu Chen, Addrian Fernandez, Omar Sinno, and Anmol Bhasin. From infrastructure to culture: A/b testing challenges in large scale social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2227–2236, 2015.

Ya Xu, Weitao Duan, and Shaochen Huang. Sqr: Balancing speed, quality and risk in online experiments. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 895–904, 2018.

Frank Yates. The analysis of multiple classifications with unequal numbers in the different classes. *Journal of the American Statistical Association*, 29(185):51–66, 1934.

Christina Lee Yu, Edoardo M Airoldi, Christian Borgs, and Jennifer T Chayes. Estimating the total treatment effect in randomized experiments with unknown network structure. *Proceedings of the National Academy of Sciences*, 119(44):e2208975119, 2022.

A Simulation Details

A.1 Under general assumptions

In Section 5.1, we compare the power of the tests given in Algorithms 1, 2 and 4.

A.1.1 Test statistics

Here, we discuss the test statistics used by the algorithms. Let $H_{i,k}$ be the fraction of treated neighbors of unit i in experiment k. Let N_i be the number of neighbors of unit i in the social network.

One experiment. For Algorithm 1, we use the following test statistic: run a linear regression of

$$Y_{\text{foc}} \sim W_{\text{foc}} + X_{\text{foc}} + N_{\text{foc}} + H_{\text{foc}}, \tag{32}$$

extract the regression coefficient of H and take the absolute value of the coefficient.

Two experiments. For Algorithm 2, we consider two different test statistics, a correlation statistic and a regression statistic. For the correlation statistic, we take

$$T(W_{\text{foc},1:2}, X_{\text{foc}}, Y_{\text{foc}}^{\text{diff}}, H_{\text{foc},1:2}) = \left| \text{Corr}\left[Y_{\text{foc}}^{\text{diff}}, H_{\text{foc},2} - H_{\text{foc},1} \right] \right|. \tag{33}$$

For the regression statistic, we run a regression of

$$Y_{\text{foc}}^{\text{diff}} \sim X_{\text{foc}} + N_{\text{foc}} + H_{\text{foc},1} + (H_{\text{foc},2} - H_{\text{foc},1}), \tag{34}$$

extract the regression coefficient of $(H_{\text{foc},2} - H_{\text{foc},1})$ and take the absolute value of the coefficient.

Three experiments. Let $T_{k,l}$ be the test statistic (regression or correlation) defined above when only two experiments are utilized (the k-th and l-th experiments are utilized). We then simply use $T_{1,2} + T_{2,3} + T_{1,3}$ as the test statistic for Algorithm 4 with $K = 3$.

A.1.2 Outcome models

We consider two different outcome models. For the linear model, let $H_{i,k}$ be the fraction of treated neighbors of unit i in experiment k. We assume

$$Y_{i,k} = (\text{signal strength})H_{i,k} + 2W_{i,k} + X_{i,1} + X_{i,2} + \varepsilon_{i,k}, \tag{35}$$

where $k \in \{1, 2, 3\}$ and $X_{i,1} \sim \mathcal{N}(0.5, 1)$, $X_{i,2} \sim \text{Poisson}(3)$ independently. The errors $\varepsilon_{i,k}$'s are such that $(\varepsilon_{i,1}, \ldots, \varepsilon_{i,K})$ is distributed as multivariate gaussian with $\mathbb{E}[\varepsilon_{i,k}] = 0$, $\text{Var}[\varepsilon_{i,k}] = 1$ and $\text{Cov}[\varepsilon_{i,k}, \varepsilon_{i,l}] = (\text{fraction of common variance})$ for $k \neq l$.

For the non-linear model, let $M_{i,k}$ be the number of treated neighbors of unit i in experiment k. We assume

$$Y_{i,k} = (\text{signal strength})\left(\frac{M_{i,k}}{20} + 5\exp\left(\frac{1}{50}\min\left(M_{i,k}, 20\right) \right) \right) + \\ 2W_{i,k} + X_{i,1} + X_{i,2} + \varepsilon_{i,k}, \tag{36}$$

where $k \in \{1, 2, 3\}$ and $X_{i,1} \sim \mathcal{N}(0.5, 1)$, $X_{i,2} \sim \text{Poisson}(3)$ independently. The errors $\varepsilon_{i,k}$'s are such that $(\varepsilon_{i,1}, \ldots, \varepsilon_{i,K})$ is distributed as multivariate gaussian with $\mathbb{E}[\varepsilon_{i,k}] = 0$, $\text{Var}[\varepsilon_{i,k}] = 1$ and $\text{Cov}[\varepsilon_{i,k}, \varepsilon_{i,l}] = (\text{fraction of common variance})$ for $k \neq l$.

A.2 Time fixed effect model

In Section 5.2, we compare the power of the tests given in Algorithms 4 and 5.

A.2.1 Test statistics

Here, we discuss the test statistics used by the algorithms. Let $H_{i,k}$ be the fraction of treated neighbors of unit i in experiment k. Let N_i be the number of neighbors of unit i in the social network.

Algorithm 4. We use the regression statistic defined in Section 5.1.

Algorithm 5. For Algorithm 5, we use an "anova" statistic. Let $\mathcal{I}_1' = \{i \in \mathcal{I}_1 : W_{i,1} = 1\}$ and let $\mathcal{I}_m' = \{m(i) : i \in \mathcal{I}_1'\}$. We start with concatenate $Y_{\text{concat}}^{\text{diff}} = \left(Y_{\mathcal{I}_1',1}^{\text{diff}}, Y_{\mathcal{I}_1,2}^{\text{diff}}, Y_{\mathcal{I}_1,3}^{\text{diff}}\right)$. Similarly, let $N_{\text{concat}} = (N_{\text{concat},1}, N_{\text{concat},m})$, where $N_{\text{concat},1} = \left(N_{\mathcal{I}_1',1}, N_{\mathcal{I}_1,2}, N_{\mathcal{I}_1,3}\right)$ and $N_{\text{concat},m} = \left(N_{\mathcal{I}_1',1}, N_{\mathcal{I}_1,2}, N_{\mathcal{I}_1,3}\right)$. We do the same concatenation for X and H. The reason we take the subset $\mathcal{I}_1'$ of $\mathcal{I}_1$ in the first experiment is that we want $Y_{\text{concat}}^{\text{diff}}$ to be a pure contrast of treatment group and control group. Without the subsetting step, Y^{diff} contains both treatment-control differences and control-control differences. Let Ind_2 be the indicator of the second experiment and Ind_3 be the indicator of the third experiment. We then run two regressions:

$$\begin{aligned}
&\text{Model 1: } Y_{\text{concat}}^{\text{diff}} \sim X_{\text{concat}} + H_{\text{concat}} + N_{\text{concat}} + \text{Ind}_2 + \text{Ind}_3, \\
&\text{Model 2: } Y_{\text{concat}}^{\text{diff}} \sim X_{\text{concat}} + N_{\text{concat}}.
\end{aligned} \tag{37}$$

Finally, we let the test statistic be the F-statistic from the anova testing of contrasting Model 1 with Model 2.

A.2.2 Matching algorithms

Random matching. We sample $m(i)$ uniformly at random without replacement.

Covariate-based matching. We use optimal matching based on the Mahalanobis distance of observed covariates and N_i [Sekhon, 2008].

A.2.3 Outcome models

We consider two different outcome models. For the linear model, let $H_{i,k}$ be the fraction of treated neighbors of unit i in experiment k. We assume

$$Y_{i,k} = (\text{signal strength})(2W_i + 1)H_{i,k} + 2W_{i,k} + X_{i,1} + X_{i,2} + \varepsilon_{i,k}, \tag{38}$$

where $k \in \{1, 2, 3\}$ and $X_{i,1} \sim \mathcal{N}(0.5, 1)$, $X_{i,2} \sim \text{Poisson}(3)$ independently. The errors $\varepsilon_{i,k}$'s are such that $(\varepsilon_{i,1}, \ldots, \varepsilon_{i,K})$ is distributed as multivariate gaussian with $\mathbb{E}[\varepsilon_{i,k}] = 0$, $\text{Var}[\varepsilon_{i,k}] = 1$ and $\text{Cov}[\varepsilon_{i,k}, \varepsilon_{i,l}] = (\text{fraction of common variance})$ for $k \neq l$.

For the non-linear model, let $M_{i,k}$ be the number of treated neighbors of unit i in experiment k. We assume

$$\begin{aligned}
Y_{i,k} = {}&(\text{signal strength})(2W_i + 1)\left(\frac{M_{i,k}}{20} + 5\exp\left(\frac{1}{50}\min\left(M_{i,k}, 20\right)\right)\right) \\
&+ 2W_{i,k} + X_{i,1}X_{i,2} + \mathbb{1}\{X_{i,1} > 0.5, X_{i,2} > 3.5\} + \varepsilon_{i,k},
\end{aligned} \tag{39}$$

where $k \in \{1, 2, 3\}$ and $X_{i,1} \sim \mathcal{N}(0.5, 1)$, $X_{i,2} \sim \text{Poisson}(3)$ independently. The errors $\varepsilon_{i,k}$'s are such that $(\varepsilon_{i,1}, \ldots, \varepsilon_{i,K})$ is distributed as multivariate gaussian with $\mathbb{E}\left[\varepsilon_{i,k}\right] = 0$, $\text{Var}\left[\varepsilon_{i,k}\right] = 1$ and $\text{Cov}\left[\varepsilon_{i,k}, \varepsilon_{i,l}\right] = (\text{fraction of common variance})$ for $k \neq l$.