Network Experimentation at Scale

Brian Karrer
Facebook
Menlo Park, CA, USA
briankarrer@fb.com

Liang Shi
Facebook
Menlo Park, CA, USA
liangshi@fb.com

Monica Bhole
Facebook
Menlo Park, CA, USA
mbhole@fb.com

Matt Goldman
Facebook
Menlo Park, CA, USA
mattgoldman@fb.com

Tyrone Palmer
Facebook
Menlo Park, CA, USA
tyronep@fb.com

Charlie Gelman
Facebook
Menlo Park, CA, USA
cgelman@fb.com

Mikael Konutgan
Facebook
Menlo Park, CA, USA
kmikael@fb.com

Feng Sun
Facebook
Menlo Park, CA, USA
sunfeng@fb.com

ABSTRACT

We describe our network experimentation framework, deployed at Facebook, which accounts for interference between experimental units. We document this system, including the design and estimation procedures, and detail insights we have gained from the many experiments that have used this system at scale. In our estimation procedure, we introduce a cluster-based regression adjustment that substantially improves precision for estimating global treatment effects, as well as a procedure to test for interference. With our regression adjustment, we find that imbalanced clusters can better account for interference than balanced clusters without sacrificing accuracy. In addition, we show that logging exposure to a treatment can result in additional variance reduction. Interference is a widely acknowledged issue in online field experiments, yet there is less evidence from real-world experiments demonstrating interference in online settings. We fill this gap by describing two case studies that capture significant network effects and highlight the value of this experimentation framework.

CCS CONCEPTS

• **Mathematics of computing** → *Exploratory data analysis.*

KEYWORDS

network interference; design of experiments; A/B testing

ACM Reference Format:
Brian Karrer, Liang Shi, Monica Bhole, Matt Goldman, Tyrone Palmer, Charlie Gelman, Mikael Konutgan, and Feng Sun. 2021. Network Experimentation at Scale. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467091

KDD '21, August 14–18, 2021, Virtual Event, Singapore.

ACM ISBN 978-1-4503-8332-5/21/08.
https://doi.org/10.1145/3447548.3467091

1 INTRODUCTION

Experimentation is ubiquitous in online services such as Facebook, LinkedIn [27], Netflix [9], etc., where the effects of product changes are explicitly tested in randomized trials. Interference, sometimes referred to as network effects, is a threat to the validity of these randomized trials as the presence of interference violates the *stable unit treatment value assumption* (SUTVA, see e.g. [6, 22, 23]) important to the analysis of these experiments. Colloquially, interference means an experimental unit's response to an intervention depends not just on their treatment, but also other units' treatments.

In this paper, we introduce a practical framework for running experiments that accounts for interference at scale and has been used for many large cluster-randomized trials at Facebook. Because the intent of these experiments is accounting for network effects, a less technical concept than interference, we refer to the framework as running *network experiments*. The framework ensures that network experiments are as easy to setup and run as any other experiment at Facebook. Our claim is not that network experiments entirely solve the problem of interference, but instead that they can be useful, scalable, general purpose tools to produce better estimates of global average treatment effects and quantify network effects. As is well known for cluster-randomized trials [13], clustering reduces statistical power, and the bias reduction can be irrelevant compared to the increased noise. To counter this, we introduce a novel cluster-based agnostic regression adjusted estimator, and our framework defaults to running side-by-side unit-randomized and cluster-randomized trials which we refer to as a *mixed experiment*, an imbalanced variation of the experimental design described by [24]. Experimenters can estimate the presence and magnitude of interference through mixed experiments, and can retroactively determine whether clustering was helpful or not.

The details of our approach and insights from running network experiments at scale can enlighten future development and methodological research. Past literature has often focused on experiments and estimation methods with size-balanced clusters (e.g.

[12, 20, 21, 24]), sometimes claimed to provide better statistical power and to avoid bias. Beyond balance being a difficult practical restriction, we find imbalanced clusters with our regression adjusted estimator can better capture interference with similar, or even superior, statistical power than balanced clusters. Agnostic regression adjustment for unit-randomized trials is described in [17], and an unbiased variant for cluster-randomized trials in [18]. We introduce another regression-adjusted estimator, suitable for our setting, that provides substantial variance reduction and differs from prior work by being biased, but we show that this bias is vanishing for experiments that use a large number of clusters. Additionally, we derive how logging which units receive treatment, so-called trigger logging [27], can be leveraged for additional variance reduction.

While there are many theoretical investigations of interference in online social networks, relatively few experiments with clear evidence of interference are described in the literature. We have found a number of experiments with apparent and substantive SUTVA violations, and we describe two such experiments in detail. To summarize, we provide:

- A framework for deploying network experiments at scale
- A detailed approach to leverage trigger logging in analysis
- An agnostic regression adjusted estimator suitable for network experiments with imbalanced clusters and achieving significant variance reduction
- Cluster evaluation indicating that imbalanced clusters are often superior in terms of bias-variance tradeoffs
- Analysis of two real-world experiments demonstrating substantial network effects

We will first review necessary background and provide further motivation in Section 2. Then we discuss our framework's implementation in Section 3, and provide our methodological contributions in Section 4. We elucidate our procedure and claims about designing clusters for network experiments in Section 5, and then apply our framework to two real-world case studies in Section 6.

2 BACKGROUND AND MOTIVATION

To make our discussion concrete, we introduce notation where random variables are capital letters and vectors are bold. Let Y_i be the random outcome for a specific experimental unit i and $W_i \in \{0, 1\}$ be the random binary treatment condition of unit i. Ideally an experiment would estimate the population mean behavior

$$\mu(\mathbf{w}) = \mathbb{E}[Y_u | \mathbf{W} = \mathbf{w}] \tag{1}$$

for an arbitrary vector of population treatment assignments $\mathbf{w}$, where the expectation corresponds to randomly selecting a unit u in the population. The global average treatment effect given by $\tau = \mu(\mathbf{1}) - \mu(\mathbf{0})$ is of primary interest. An ideal experiment provides an understanding of *what would happen to a random unit when all units are placed in one condition versus another condition.* SUTVA implies that $\mu(\mathbf{w}) = \mathbb{E}[Y_u | W_u = w_u]$, and hence $\tau = \mathbb{E}[Y_u | W_u = 1] - \mathbb{E}[Y_u | W_u = 0]$. Fundamentally, SUTVA implies symmetries in the function μ, and these symmetries are leveraged to estimate τ from experiments.

A typical unit-randomized test (AB test) that treats a random fraction p of the units and ignores interference estimates

$$\tau_{unit}(p) = \mathbb{E}[Y_u | W_u = 1, p] - \mathbb{E}[Y_u | W_u = 0, p], \tag{2}$$

where the presence of p reminds us that this expectation also includes randomizing this fraction of other units into treatment. This only equals τ if SUTVA holds, and can be quite different otherwise.

Depending on the treatment that is tested, as well as the outcomes of interest, SUTVA may not hold. Cluster-randomized trials are an approach to better account for interference among experimental units, distinguished from an AB test through randomizing clusters of units to conditions. Assume the population is partitioned into clusters where c_u is the cluster assignment for unit u. If a random fraction p of clusters are treated, then a typical cluster-randomized test would estimate

$$\tau_{cluster}(p) = \mathbb{E}[Y_u | \mathbf{W}_{c_u} = \mathbf{1}, p] - \mathbb{E}[Y_u | \mathbf{W}_{c_u} = \mathbf{0}, p] \tag{3}$$

where the notation $\mathbf{W}_{c_u}$ means the vector of treatment assignments restricted to cluster c_u. This estimand may have less bias compared to the true τ than τ_{unit} since the random unit is placed into an experience "closer" to the counterfactual situation of interest [10]. In order for $\tau_{cluster}$ to equal τ, we require no interference between clusters and can have arbitrary interference within clusters. This situation is referred to as *partial interference* [15] and should be viewed as SUTVA for clusters.

Relevant clusters that capture interference are often unclear. When spatial interactions are suspected as the cause of interference, geographic clusters, such as zip codes or Google's GeoCUTS [21] might be appropriate. Within an online social network, interactions between users are important to the user experience and often a suspected cause of interference. These interactions can be logged and viewed as a weighted graph correlated with interference. Clustering this graph and randomizing the resulting clusters has been proposed as graph-cluster randomization [24, 25].

The degree to which clusters capture relevant interference is an empirical question that can be confronted with experimental data. A variety of hypothesis tests have been devised for the presence of interference or network effects, with or without clusters. The general idea is that a lack of interference implies equality across different estimands, which each can be estimated from the same experiment. On the analysis side, exposure modeling [1] utilizes importance sampling through reweighing results. Another approach splits experimental units into focal and other units and estimates whether variation in the other units' treatments affects the focal units [2]. On the design side, [24] advocate an experimental design with balanced clusters that runs a side-by-side unit and cluster-randomized trial, where significant differences indicates the presence of interference. An imbalanced variant was used by [14], and our framework provides another imbalanced version of this mixed experimental design by default.

3 NETWORK EXPERIMENT IMPLEMENTATION

Our implementation has two primary components that deploy treatments and manage clusterings respectively. The cluster management details are described in the supplement. The component that deploys treatments is depicted visually in Figure 1, where the figure should be read from left to right. A clustering of experimental units, represented by larger circles encompassing colored dots for units,

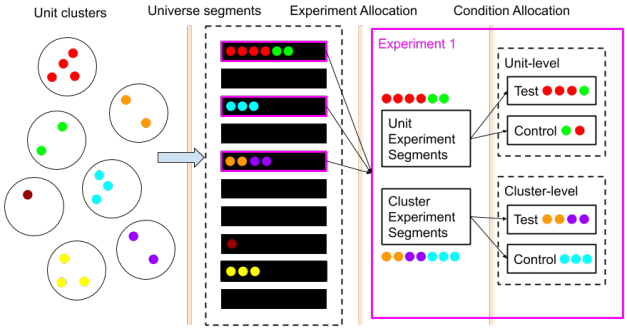

Figure 1: Visualization of the network experiment randomization process.

is taken as input. A given clustering and the associated units define a *universe*, the population under consideration.

These clusters of experimental units are hashed into *universe segments* based on the universe name, which are then allocated to experiments. Universe segments allow a universe to contain multiple mutually exclusive experiments at any given time, a requirement for a system used by engineering teams. After allocation to an experiment, segments are randomly split via a deterministic hash based on the experiment name into unit-randomized segments and cluster-randomized segments. The final condition allocation hashes units or clusters into treatment conditions, depending on whether the segment has been allocated to unit or cluster-randomization. The result produces the $\mathbf{W}$ treatment vector that is used for the experiment. For convenience, we denote $R_i = 1$ if unit i was cluster-randomized and $R_i = 0$ if unit i was unit-randomized.

3.1 Trigger logging

In practice, the allocation W_i for a unit i is only computed upon a call to the service supporting network experiments. This call includes the universe name and experiment name, and a distributed key-value store is queried for the appropriate cluster of unit i for that experiment. If unit i does not have a cluster or their cluster is not assigned to the experiment, they are excluded from the experiment's population, which in most cases means they receive the default control treatment. Units in the experiment that call the service are logged along with their treatment assignment, which we follow [27] in referring to as *triggering* their assignment.

Trigger logging implies invariances for cluster-randomized experiments that we show can be leveraged in analysis. We let $T_i = 1$ if unit i has a treatment assignment logged by the network experiment service (i.e. the unit is triggered) and $T_i = 0$ otherwise. A unit's observed outcome cannot causally depend on their own treatment assignment unless they are triggered, as this treatment assignment is never used to do anything for that unit. Hence $\mathbb{E}[Y_i|\mathbf{W}_{-i}, W_i = x, T_i = 0]$ and $P(T_i = 0|\mathbf{W}_{-i}, W_i = x)$ do not depend on x, where $\mathbf{W}_{-i}$ means the vector of treatment assignments excluding unit i. An analogous statement in terms of a cluster as a whole can be phrased. Let the notation $\mathbf{T}_{c_i} = \mathbf{0}$ mean no unit in the cluster of unit i is triggered. We refer to a cluster with $\mathbf{T}_{c_i} \neq \mathbf{0}$ as a triggered cluster. Then we have that $\mathbb{E}[Y_i|\mathbf{W}_{-c_i}, \mathbf{W}_{c_i} = \mathbf{x}, \mathbf{T}_{c_i} = \mathbf{0}]$ and $P(\mathbf{T}_{c_i} = \mathbf{0}|\mathbf{W}_{-c_i}, \mathbf{W}_{c_i} = \mathbf{x})$ do not depend on $\mathbf{x}$.

4 NETWORK EXPERIMENT ANALYSIS

Before describing our estimation strategy, we describe how we identify treatment effects from our experimental design.

4.1 Basic estimand

Our basic estimand from which we will construct other estimands is

$$\mu(w, r) = \mathbb{E}[Y_u|W_u = w, R_u = r]. \tag{4}$$

This estimand answers the counterfactual question of *what is the expected outcome of a random unit placed into condition w under cluster ($r = 1$) or unit ($r = 0$) randomization under the experimental design.* It is important to understand that there are two, and possibly three, aspects of randomness to the above expectation. The first is the experimental design's allocation of the remaining population to conditions. The second is the selection of the random unit in the population. If we wish to refer to a specific unit i in the population, instead of a randomly selected unit u, we will write Y_i. Lastly, we assume

Assumption 1. *Every outcome Y_i, including T_i, is deterministic given treatment assignments $\mathbf{W}$ for all units i.*

A unit sampled into our experiment through unit-randomization is equivalent to a random unit in the population, so averages suffice to identify this estimand for $r = 0$. A unit sampled via cluster-randomization is **not** a random unit in the population because this requires sampling clusters proportional to cluster size. We can utilize importance sampling based on cluster size to adjust for this. Letting S_c represent the size of a cluster c and $\mathbb{E}[]$ indicate the average over a randomly selected cluster c in the population, the basic estimand equals

$$\mu(w, 1) = \frac{\mathbb{E}[Y_c|W_c = w]}{\mathbb{E}[S_c]}, \tag{5}$$

where Y_c is the sum over outcomes for all units in cluster c and $W_c \in \{0, 1\}$ is the treatment assignment of cluster c.

We can then construct standard contrasts between conditions, including a difference-in-means estimand $\mu(1, r) - \mu(0, r)$ and a ratio estimand of $\frac{\mu(1,r)}{\mu(0,r)} - 1$. Importantly, we can also consider contrasts across unit- and cluster-randomized conditions which allows for testing if unit-level SUTVA holds [24]. If SUTVA is true, then in Eq. 4 the allocation of the remaining population is irrelevant. Further, whether a unit was cluster-randomized or unit-randomized as indicated by r is also irrelevant. So SUTVA would imply that $\mu(w, 1) - \mu(w, 0)$ is zero. Beyond hypothesis testing, the magnitude of the difference is informative of the magnitude of interference.

4.2 Leveraging trigger logging

Trigger logging allows more precise inferences if we make assumptions about interference and trigger logging. We consider the following two assumptions:

Assumption 2. ***(SUTVA for triggering)*** *The outcome T_i does not depend on any unit's treatment assignments, i.e., $P(T_i = 0|\mathbf{W}) = P(T_i = 0)$ for all $\mathbf{W}$ and i.*

Assumption 3. ***(Conditional SUTVA)*** *The outcome Y_i, conditioned on $T_i = 0$, does not depend on any unit's treatment assignments, i.e., $\mathbb{E}[Y_i|\mathbf{W}, T_i = 0] = \mathbb{E}[Y_i|T_i = 0]$ for all $\mathbf{W}$ and i.*

Applying all three assumptions, $\mu(w, r)$ becomes

$$P(T_u = 1)\,\mathbb{E}[Y_u|W_u = w, R_u = r, T_u = 1] + P(T_u = 0)\,\mathbb{E}[Y_u|T_u = 0].$$

The second term is not a function of w or r, and therefore cancels out of contrasts across conditions.[1] Hence under these assumptions, we only need to estimate

$$\mathbb{E}[Y_u|W_u = w, R_u = r, T_u = 1],$$

because contrasts will not contain contributions from non-triggered units.[2] Filtering the experiment's population to triggered units can result in large gains in precision since terms known to be zero are excluded from the contrasts. However, those terms are only zero if the assumptions hold.

We use two tests to check these assumptions. First, we test SUTVA violations in triggering via examining if $\mu(w, r)$ depends on r using triggering as the outcome. In practice, we compare the average number of triggered units per triggered cluster of different cluster-randomized conditions, which should be equal if SUTVA holds for triggering. If this test passes, we check for violations of conditional SUTVA, by comparing an estimate for the outcome Y of $\mu(1, 1)$ to $\mu(0, 1)$ limited to untriggered users in triggered clusters.

If these tests pass, we condition our analysis on just the population of triggered units. If both tests do not pass, we drop both assumptions, focus on $r = 1$, and instead assume SUTVA at the cluster-level for triggering such that $P(\mathbf{T}_c|\mathbf{W}) = P(\mathbf{T}_c|\mathbf{W}_c)$ for any cluster c. Combined with the discussion in Section 3.1, this implies $P(\mathbf{T}_c = \mathbf{0}|\mathbf{W}) = P(\mathbf{T}_c = \mathbf{0})$ and also that $\mathbb{E}[Y_u|W_u = w, R_u = 1, \mathbf{T}_{c_u} = \mathbf{0}] = \mathbb{E}[Y_u|\mathbf{T}_{c_u} = \mathbf{0}]$. Under these assumptions, we have

$$\begin{aligned}\mu(w, 1) = &\big(P(\mathbf{T}_{c_u} \neq \mathbf{0})\,\mathbb{E}[Y_u|W_u = w, R_u = 1, \mathbf{T}_{c_u} \neq \mathbf{0}] + \\ &P(\mathbf{T}_{c_u} = \mathbf{0})\,\mathbb{E}[Y_u|\mathbf{T}_{c_u} = \mathbf{0}]\big).\end{aligned}$$

If partial interference for triggering does not hold, our estimates will contain additional bias, but most of our network experiments did not even provide evidence for unit-level interference in triggering.

4.3 Estimation

Estimation for cluster-randomized trials can proceed via cluster-level summaries, mixed effect models, or generalized estimating equations [13]. For network experiments, we favor an approach that is simple to implement at scale and explain.

Our estimands can be estimated via sample averages. In particular, we perform a conditional inference that views our experimental design as providing $k(w, r)$ samples of units or clusters with $W = w$ and $R = r$.[3] Units in unit-randomized conditions ($r = 0$) can be considered as clusters of size 1. If SUTVA for triggering and conditional SUTVA for Y are deemed valid for the experiment, these clusters just consist of triggered units. Each cluster is accompanied by (Y, W, R, X, S), where Y are experimental outcomes summed across the cluster, W and R define the treatment condition, X are summed pre-treatment covariates, and S is the cluster size.

Let $\overline{A}(w, r)$ be the sample mean of Y, X, or S, or functions of these variables, across all clusters with $W = w$ and $R = r$ sampled into the experiment as follows

$$\overline{A}(w, r) = \frac{1}{k(w, r)} \sum_c 1(W_c = w, R_c = r) A_c.$$

For simplicity, we will hereafter omit the (w, r) for any quantity that are defined on this group of clusters and only add this notation as needed for clarity. An estimator for estimand μ in Eq. 4 is then

$$\widehat{\mu} = \frac{\overline{Y}}{\overline{S}}.$$

This estimator is asymptotically consistent, in that as the number of clusters k goes to infinity, it equals our estimand [18]. For a finite number of clusters, the estimator is biased for $r = 1$ because the average cluster size is random.

To understand properties of this estimator, we apply the delta method (see [7] for a recent discussion) to second-order and derive the bias as

$$\mathbb{E}\left[\widehat{\mu} - \mu\right] \approx \frac{1}{\mathbb{E}[S]^2}\left(\mu Var(\overline{S}) - Cov(\overline{Y}, \overline{S})\right),$$

where $\mathbb{E}[S]$ is the population mean of S. Note that this bias is zero if all clusters have the same size, a preference expressed in [12]. This requirement is unnecessary though under the following:

Assumption 4. *Covariances of any two sample means, such as $Cov(\overline{Y}, \overline{S})$, under our experimental design are $O(1/k)$.*

Assuming SUTVA is sufficient but not necessary for this to happen, as including limited dependence between variables could still produce this asymptotic behavior [4]. Of course, we should keep in mind that with too strong of dependence, this asymptotic behavior may be incorrect. With this caveat, the delta method shows the bias is $O(1/k)$ when $r = 1$ and zero otherwise.

Using the first-order term in the delta method expansion, we have

$$Var(\widehat{\mu}) \approx \frac{1}{\mathbb{E}[S]^2} Var\left(\overline{Y} - \mu\overline{S}\right)$$

and hence the variance is also $O(1/k)$ under our assumption. The root mean-squared error is then dominated by the standard deviation of order $O(1/\sqrt{k})$. A benefit of online experimentation is the experiments have a very large number of clusters, so we can consider any terms, like the bias, of order $O(1/k)$ as irrelevant.

At this point, one might consider the estimation largely complete. We can compute $\widehat{\mu}$, and confidence intervals with additional assumptions, from the experimental data. A difficulty with this approach is that the resulting confidence intervals can be very large, especially for contrasts between the unit-randomized and cluster-randomized conditions as typically $k(w, 1) \ll k(w, 0)$. Variance reduction is required to produce reasonably-sized confidence intervals.

4.4 Agnostic regression adjustment

We achieve this variance reduction through introducing cluster-based regression adjustment. We can define a collection of many sample means from the experimental data. Unfortunately, we do

[1] The conditional SUTVA assumption is only needed for $r = 1$ to be included in this statement. The second term for $r = 0$ would be $\mathbb{E}[Y_u|T_u = 0]$ without this assumption.

[2] Strictly speaking, this is untrue for the ratio estimand where the denominator should not condition on $T_u = 1$. In this case, we redefine to include the conditioning.

[3] The rest of the population is assumed to be in control for the purpose of interference.

not know population means corresponding to these sample means, but we do have that, for some quantities (e.g. the pre-treatment covariates X), the contrast across conditions (w, r) and (w', r') is asymptotically vanishing

$$\mathbb{E}[\widehat{\mu_X} - \widehat{\mu_X}'] \approx O(1/k) \approx 0,$$

where $\widehat{\mu_X}$ and $\widehat{\mu_X}'$ are simply $\frac{\overline{X}}{\overline{S}}$ for the conditions in the contrast. Let ϕ be a vector of such quantities (or linear combinations such as $\sum_{w,r} \beta_{wr} \widehat{\mu_X}(w, r)$ where β_{wr} sums to zero), where $\mathbb{E}[\phi]$ is $O(1/k)$ and $Var(\phi)$ is $O(1/k)$.

Then our regression-adjusted estimator is defined as a function of a vector γ with the same length as ϕ

$$\widehat{\mu_\gamma} = \widehat{\mu} - \gamma\phi.$$

This adjusted estimator has bias $O(1/k)$ and the variance considered as a function of γ is

$$Var(\widehat{\mu_\gamma}) = Var(\widehat{\mu}) - 2\gamma Cov(\phi, \widehat{\mu}) + \gamma Var(\phi)\gamma$$

with a minimum when

$$\gamma^* = Var(\phi)^{-1} Cov(\phi, \widehat{\mu}).$$

The trick for regression adjustment is to estimate a $\widehat{\gamma}$ from experimental data (ideally γ^*). Let $\widehat{\gamma}$ be a consistent estimator of some γ that has bias $O(1/k)$ and variance $O(1/k)$. We then have that

$$\widehat{\mu_{\widehat{\gamma}}} - \widehat{\mu_\gamma} = -(\widehat{\gamma} - \gamma)\phi.$$

This is the dot product of two terms of order $O(1/\sqrt{k})$ and (assuming a constant number of terms) is therefore $O(1/k)$. The additional bias due to regression adjustment is therefore irrelevant, and we simply ignore that we estimated γ such that

$$Var(\widehat{\mu_{\widehat{\gamma}}}) \approx Var(\widehat{\mu}) - 2\widehat{\gamma} Cov(\phi, \widehat{\mu}) + \widehat{\gamma} Var(\phi)\widehat{\gamma}.$$

Estimating γ^* directly appears difficult, so we apply the delta method to the covariances and variances in γ^*, and estimate both expanded expressions instead. This will match γ^* asymptotically, assuming the estimator is consistent and the delta method is valid.[4]

In practice for ϕ, we utilize the scalar pre-experiment value of the outcome of interest for the two conditions in a contrast. This single-variable regression adjustment when applied to unit-randomized conditions has been referred to as CUPED [8, 26]. Our regression-adjusted estimator between (w, r) and (w', r') for difference-in-means is $\widehat{\mu_{\widehat{\gamma}}} - \widehat{\mu_{\widehat{\gamma}}}'$, and similarly, $\widehat{\mu_{\widehat{\gamma}}}/\widehat{\mu_{\widehat{\gamma}}}' - 1$ for the ratio estimand. For both estimators, we utilize the delta method to derive variance expressions, treating $\widehat{\gamma}$ terms as fixed non-random quantities.

All of these tedious-to-derive variance expressions for the estimators (and for $\widehat{\gamma}$) are in terms of covariances and variances of sample averages, such as $Var(\overline{Y})$, $Cov(\overline{S}, \overline{Y})$, $Cov(\overline{Y}, \overline{X})$, as well as population means. To estimate these expressions, we substitute in sample averages for the population means, but the covariances require additional assumptions.

Assumption 5. *We assume an infinite population of clusters to ignore covariances across conditions.*

[4] Alternatively, one could estimate a reasonable γ using weighted linear regression or even ordinary least squares. The resulting expression will be different from the delta method estimator we utilize for γ though. Any reasonable estimating equation for γ that has the appropriate asymptotic behavior should result in variance reduction.

Under this assumption, we can estimate sample mean covariances that do not include an outcome Y via empirical covariances

$$\widehat{Cov}(\overline{A}, \overline{B'}) = \begin{cases} \frac{\widehat{Cov}(A, B')}{k} & \text{if } w = w', r = r' \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

So far we have only assumed asymptotic properties that can hold without SUTVA.

Assumption 6. *For estimating covariances involving outcomes Y, we assume the appropriate form of SUTVA holds (unit-level for $r = 0$ and cluster-level for $r = 1$) such that the above expression is valid.*

Therefore unlike our point estimates, our asymptotic standard errors and associated confidence intervals rely upon SUTVA.

5 CLUSTER DESIGN

A good clustering of experiment units is important for network experiments, and needs to be considered in the context of the implementation and analysis. An ideal clustering includes all interference within clusters, so there is no interference between clusters, which removes any estimand bias. To be more specific, we consider the setting of graph-cluster randomization where a domain expert has created a relevant graph G where each unit in the population is a vertex and edges, possibly weighted, represent a hypothetical interference pattern among the units. The creation of the graph requires strong domain knowledge, but we do not require that this graph precisely represent interference. A higher quality graph will just lead to higher quality clusters.

Let purity denote the fraction of edges within clusters. We assume a clustering with higher purity has less bias. A naive approach that achieves 100% purity, and captures all interference, groups all units into a giant single cluster. This is obviously unacceptable for a cluster-randomized experiment and so purity is just one aspect of cluster quality. An experiment should also have enough statistical power to detect treatment effects. A single cluster including all units has no power, and a clustering that puts every unit in its own cluster, equivalent to unit randomization, has good power but the worst purity (zero). The tradeoff between purity and power is a *bias-variance tradeoff*: higher purity leads to less bias while more statistical power requires smaller clusters.

We consider two prototypical clustering algorithms: Louvain community detection (Louvain) [3] and recursive balanced partitioning (BP) [16]. We chose these algorithms because they have extremely scalable implementations. Importantly, these algorithms can produce a very large number of clusters from graphs representing large populations, which was shown to be important for sufficiently low variance estimation in Section 4.3. Moreover, the two algorithms are emblematic of clustering approaches that produce imbalanced and balanced cluster sizes. Louvain generally produces very imbalanced clusters with a heavy-tailed cluster-size distribution, whereas BP generates balanced clusters.

We find that *imbalanced graph clusters are typically superior in terms of the bias-variance tradeoff for graph-cluster randomization.* Other clustering algorithms may have different tradeoffs and can be evaluated via the tools we provide. While an extensive comparison across algorithms is outside the scope of this paper, we suggest that

imbalanced clustering algorithms, like Louvain, always be considered. To demonstrate this, we describe our evaluation procedure.

5.1 Evaluating the bias-variance tradeoff

To evaluate a clustering's statistical power for experiments, we run synthetic Monte-Carlo AA tests assuming partial interference holds for the given clustering and apply the estimation procedure described in Section 4.3. If similar experiments have been run, we can reuse trigger logging from these past experiments, otherwise we assume a random fraction of the population is triggered on each simulation. For outcome Y, we consider the primary metric of interest and let X be pre-synthetic experiment values of the same metric to enable cluster-based regression adjustment.

After validating that statistical properties of the estimator are reasonable (i.e. coverage, CI width, etc.)[5], we produce an estimate of the minimal detectable effect (MDE), a simple transformation of the estimator standard deviation. A larger standard deviation means a larger MDE. We represent our bias-variance tradeoff by plotting the MDE-purity tradeoff. An ideal clustering would have high purity and low MDE.

We describe results for a particular clustering, but we emphasize that these results are qualitatively similar to our general experience designing clusters for graph-cluster randomization. We applied Louvain and BP with various parameters on a graph consisting of about 90 billion edges and about 2 billion vertices. Louvain optimizes modularity [19], a quantity related to purity. Louvain includes a resolution parameter, and in our implementation, smaller resolution leads to smaller clusters. We consider $\{0.0001, 0.00001\}$ for the resolution parameter. Our Louvain implementation repeatedly iterates on a clustering across iterations, and we consider clusters produced at iterations 3 and 5. A single run of BP also provides multiple clusterings through recursively generating a binary tree. Each level of the binary tree corresponds to a partition of units, where level k contains 2^k clusters. We consider BP clusters produced by levels 17 and 20. As shown in the supplement, cluster size distributions are highly skewed for Louvain and balanced for BP.

For each clustering, we estimate the MDE at a given power (95%). The tradeoff between MDE and purity is shown in Fig. 2: for either algorithm, a higher purity is often associated with a larger MDE. The two algorithms perform very differently in terms of purity and MDE. Specifically, Louvain generates imbalanced clusterings with much higher purity but also comparable MDE to BP, providing evidence against claims that balanced clusters should be preferred for graph-cluster randomization.

6 NETWORK EXPERIMENT CASE STUDIES

We now describe two network experiments in detail, using graph and geographic clusterings respectively.

6.1 Stories experiment

The Facebook Stories Viewer Experience team conducted a two-weeks-long mixed network experiment for Android and iOS users on a new reply design: in the control group, when users view a Facebook story they would see a horizontal scroll bar (left in Fig. 3)

[5]The estimation procedure can perform poorly if outlier clusters contain a non-trivial fraction of all units in the population.

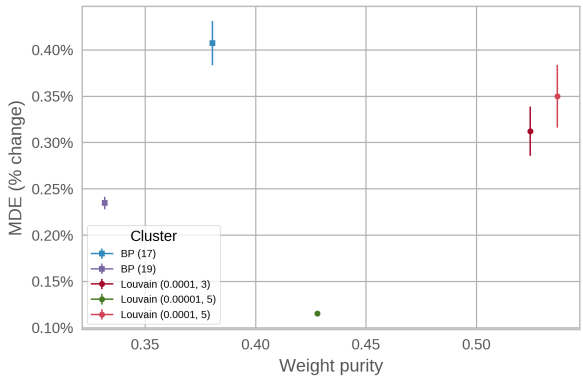

Figure 2: MDE vs purity for Louvain and recursive BP with different parameters: BP (x) is BP with level x, and Louvain (a, b) is Louvain with resolution a and iteration b.

with all emojis and in the test group users saw either a blue thumb-up (Android, right in Fig. 3) or a transparent heart (iOS, middle Fig. 3) next to the text reply box. All emojis would pop up after clicking on the thumb-up or heart.

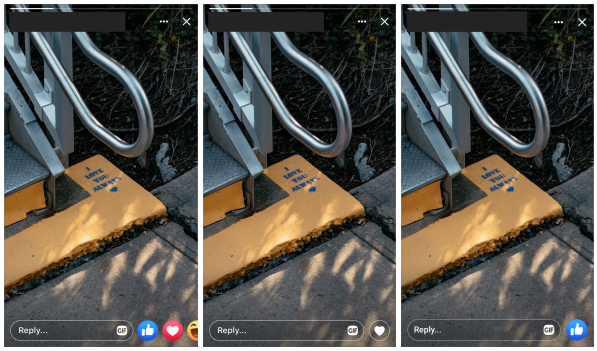

Figure 3: Test and control in Stories experiment. (left) Control group: all emojis can be seen by horizontally scrolling the bottom bar; (middle) Test group for iOS: static transparent heart; (right) Test group for Android: static blue thumb-up. To access all emojis in the test group we need to click on either the heart or the thumb-up.

This feature can alter a user's experience as both a story *viewer* and as a story *creator*. Previous user-randomized experiments indicated that the new feature increased text replies sent to stories and decreased the usage and send rate of emojis, but did not detect an effect on story creators, since creators received feedback from viewers in both the test and control groups. To estimate the indirect treatment effects on users who create stories, we ran a mixed experiment expected to reduce the interference between test and control viewers for story creators and estimate the presence and magnitude of network effects. Users were grouped into clusters using the Louvain algorithm based on historical interaction data for Facebook Stories.

Table 1: Estimated ATE for different contrasts in the Stories mixed experiment on triggered subpopulation

Metric	cluster test - cluster ctrl	user test - user ctrl	cluster test - user test	cluster ctrl - user ctrl
Story Viewer Metrics				
emoji replies given	−23.84% ± 0.69%	−23.62% ± 0.16%	−0.19% ± 0.36%	0.20% ± 0.35%
text replies given	11.06% ± 0.56%	11.76% ± 0.17%	−0.57% ± 0.36%	−0.06% ± 0.34%
Story Creator Metrics				
emoji replies received	−6.37% ± 0.87%	0.73% ± 2.44%	−5.90% ± 2.38%	1.40% ± 0.73%
text replies received	2.01% ± 0.56%	−0.13% ± 0.26%	1.69% ± 0.44%	−0.52% ± 0.44%
percent of creators with feedback	−0.91% ± 0.07%	−0.05% ± 0.04%	−0.78% ± 0.06%	0.10% ± 0.05%
number of creators with feedback	−1.16% ± 0.21%	−0.21% ± 0.08%	−0.96% ± 0.16%	0.00% ± 0.16%
daily active creators	−0.25% ± 0.18%	−0.17% ± 0.07%	−0.18% ± 0.14%	−0.09% ± 0.14%

6.1.1 Hypothesis. The metrics of interests include: emoji replies from story viewers, text replies from viewers, emoji replies received by story creators, text replies received by story creators, daily active story creators, number of story creators with at least one feedback and percentage of creators receiving at least one feedback.

While the hypothesis is that the new experience would directly increase text replies and decrease lightweight replies from viewers, the purpose of the mixed experiment to measure the downstream treatment effects on story creator metrics. Since cluster-randomization reduces test-control interference on story creators from viewer replies, we expect to measure larger differences between the user- and cluster-side estimates for story creator metrics than story viewer metrics.

6.1.2 Results. Point estimates and 95% confidence intervals of the ratio estimators for cluster-randomized conditions are shown in Table 1 in the column "cluster test - cluster ctrl". The results for text and emoji replies are consistent with our expectations: emoji replies decreased by 23.8% while the text replies increased by 11.1%. The movement in replies received by story creators are consistent with the replies sent by viewers, though the effect sixes are smaller: -6.4% vs -23.8% for emoji replies and 2% vs 11% for text replies received and sent, respectively. Since story creators still receive replies from test and control viewers (the cluster purity is about 40%), this difference is expected. The cluster-randomized experiment also shows a 0.25% drop in daily active creators and a 1.2% drop in the number of active creators with feedback, resulting in in a 0.9% drop in the percent of story creators with feedback.

As expected, story creator metrics in the user-side are not significant and smaller than in the cluster-side, due to large test-control interference from viewers, which provides evidence on the effectiveness of cluster-randomization to reduce interference. To quantify the difference in user- and cluster-side estimates, we directly compare the cluster-randomized and user-randomized test groups in the last two columns of Table 1. The estimates of the cluster test group for all story creator metrics are significantly different from those of the user test group. By design, producers in the cluster test group receive replies more often from viewers within the same cluster, hence receiving less emoji replies and more text replies, resulting in less replies overall, thus decreasing daily active creators. The same applies for the number of story creators with feedback, yet compounded with the effects on the viewer reply metrics.

Overall, these estimates are consistent with our hypotheses, and the large differences in story creator metrics relative to the overall treatment effects provides evidence that network experiments capture more interference than previous user-randomized designs and lead to more reasonable estimates.

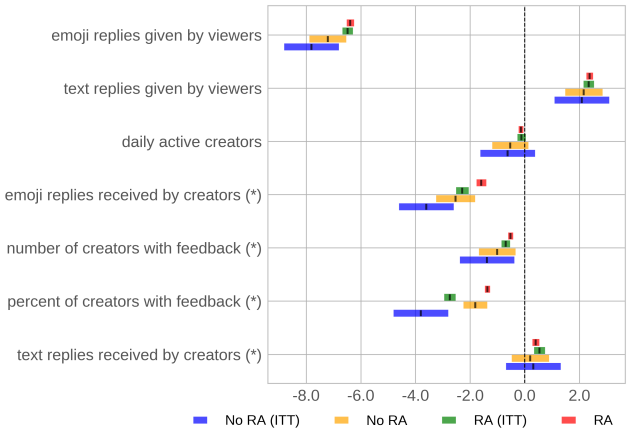

Figure 4: Comparison of ATE estimates with scaled 95% confidence intervals computed on triggered users and triggered clusters (ITT), with and without regression adjustment (RA) for cluster test versus cluster control in the Stories experiment. The story creator metrics labeled with (*) do not pass the conditional SUTVA check.

6.1.3 Methodology variations. So far, we have presented results on the sub-population of triggered users using regression adjustment. In Figure 4, we show how results for the cluster test and cluster control comparison change if we perform an Intent-To-Treat (ITT) analysis on the triggered clusters as described in Section 4.2, instead of triggered users, and if we do not use regression adjustment. Each metric row was re-scaled such that the ITT estimate without regression adjustment has a confidence interval (CI) width of one.

We clearly see that regression adjustment (RA) provides substantial precision gains. Conditioning on the triggered users provides additional gains, although of smaller magnitude. Looking closely at the comparison between RA (ITT) and RA, we do see significant

Table 2: Commuting Zones experiment results

Metric	Unit of Randomization	Estimated Effect (with 95% CI)
Applications to jobs with no previous applications	user	71.841% ± 5.087%
Applications to jobs with no previous applications	cluster	49.652% ± 17.817%
Probability a job receives an application	cluster	14.069% ± 11.198%
Probability an employer posted a new job	cluster	16.959% ± 15.731%

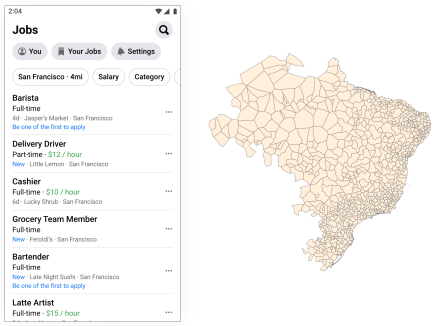

Figure 5: An example of Facebook's job board (left) and Commuting Zones in Brazil (right).

differences for some metrics. Indeed, the metrics labeled by a star are ones for which RA (ITT) may be more accurate. For this experiment, we concluded that SUTVA for triggering was valid, but our testing for conditional SUTVA for Y depended on the metric of interest. Indeed, most story creator side metrics, labeled with a star, failed this conditional SUTVA test, and we see evidence of this in the different estimates provided by RA and RA (ITT).

We note though that while the ITT estimates may be mildly different on the starred creator metrics, the qualitative interpretation of the results on the triggered clusters remains consistent with the results on the triggered users presented in the previous subsection.

6.2 Commuting Zones experiment

Given the nature of Facebook, a natural way to cluster individuals might be through friendship networks or online engagements between content producers and consumers, as in the previous subsection. Sometimes, a relevant clustering can be defined by geographical location. At Facebook this might be particularly the case for products like Jobs on Facebook (JoF), where individuals are likely to interact with employers closer to their own location.

We describe an example of an experiment for JoF using geographical clusters. JoF connects hiring businesses to FB users by enabling employers to list jobs to which users can apply. Since this is a two-sided market, changes that affect user behavior may impact employer behavior, and isolating these total effects is difficult in a user-randomized experiment.

The JoF team wanted to understand the overall ecosystem effects of up-ranking jobs that had few previous applications at the time a user was on the job board. Connecting users to jobs with fewer applications can help users and employers by connecting users to employers who are most likely to hire them, and by improving the experience for employers with few previous engagements (who might get the highest marginal utility from an application). This may lead employers to post more jobs, allowing the platform to provide more opportunities to future job seekers.

In a user test, the team boosted jobs with few previous applications in the job board and tracked applications sent on the platform as well as applications sent to jobs that had no previous applications at the time of submission. This second metric served as a proxy for the impact on creators, since it is difficult to measure this impact directly in a user-randomized experiment. One issue with this metric is the interference between the treatment and control groups. By boosting jobs without previous applications, we increase the visibility of these jobs in the treatment group which leads more users to apply to these jobs. When users in the control group apply, these jobs now have one application or more, but some of these users would have applied to the job anyway. This interference causes a user-test to overstate the treatment effect on applications to jobs with no previous applications.

Since users search for jobs in their general location, the team ran this experiment using Facebook commuting zones, a clustering available through Facebook's Data for Good program, to account for interference between users in the treatment and control groups [11]. Figure 5 shows an example of the JoF job board on the left, and of commuting zones in Brazil on the right.

6.2.1 Results. The team ran a mixed experiment where trigger logging was useful since a small fraction of the population is looking for new jobs at any given time. Table 2 summarizes the results of this test from a 2.5 week period. In the user-side results, the metric of interest (applications to jobs with no previous applications) increased by 71.8%, consistent with a previous user-test. The commuting zone mixed experiment, however, showed that the user-side treatment effects were upwardly biased. The cluster-randomized estimate was instead a 49.7% increase, a difference that is statistically significant at the 5% level. This comparison benefited substantially from regression adjustment, which can reduce the size of the standard errors in commuting zone experiments by over 30%. In the absence of these results the team might have inferred they were increasing applications to these jobs more than was truly the case.

By randomizing this experiment at the commuting zone (geographical) level, the team confirmed a number of hypotheses. First, user-randomization leads to significant bias in the metric of applications to jobs with zero previous applications. Second, as indicated in Table 2 and described in the supplement, changes to the user

experience that increase this metric do in fact cause employers to post more jobs on the platform (the probability that an employer posted another job increased 17.0%). Understanding the interactions between applicants and employers in a two-sided marketplace is important for the health of such a marketplace, through network experiments we can better understand these interactions.

7 DISCUSSION

We have introduced a practical framework for designing, implementing, and analyzing network experiments at scale. Our implementation accommodates mixed experiments, cluster updates, and the need to support multiple concurrent experiments. Our analysis procedure results in substantial variance reduction by leveraging trigger logging as well as our novel cluster-based regression adjusted estimator. We also introduce a procedure that allows researchers to evaluate bias-variance tradeoffs for clustering methods, and show that the tradeoffs are often in favor of imbalanced clusters.

The two mixed experiments, one leveraging graph-cluster randomization and another geographic clusters, demonstrate the flexibility and value of network experiments. In both cases, with the help of large precision gains from regression adjustment, we estimated significant interference effects compatible with our hypotheses, with clear evidence of differences between cluster- and user-randomized conditions. These effects had been hidden in prior user-randomized experiments testing the same changes, indicating that network experiment results are not merely differences in magnitude, but can provide differences in interpretation.

These two experiments were chosen by the authors for the purpose of illustration. Not all tests benefit from cluster-randomization or show significant network effects in a mixed experiment, especially if interference is weak or non-existent relative to the direct treatment effects. Our expectation is that many product changes may not require a network experiment for accurate estimates of the global average treatment effect. Under weak interference, the bias-variance tradeoff favors user-randomized designs. That said, we believe interference is often ignored for the purpose of convenience and not due to prior knowledge. Through our framework, experimenters can learn whether interference is relevant in their domain by easily running a network experiment. On the flip side of ignoring interference, we also have observed claims that interference would have led to improved results relative to those in user-randomized experiments. With our framework we can now test such claims about the direction and magnitude of interference.

Our design-based results are only unbiased under partial interference, and we view network experiments as an approach that trades bias for robustness, simplicity, scale, and convenience. One compelling avenue for future research is to consider alternative estimation procedures for data generated from network experiments that carefully applies additional modeling assumptions.

REFERENCES

[1] Peter M. Aronow and Cyrus Samii. 2017. Estimating average causal effects under general interference, with application to a social network experiment. *Ann. Appl. Stat.* 11, 4 (12 2017), 1912–1947.

[2] Susan Athey, Dean Eckles, and Guido W. Imbens. 2018. Exact p-Values for Network Interference. *J. Amer. Statist. Assoc.* 113, 521 (2018), 230–240.

[3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[4] Louis H. Y. Chen and Qi-Man Shao. 2004. Normal approximation under local dependence. *Ann. Probab.* 32, 3 (07 2004), 1985–2028.

[5] Alex Chin. 2019. Regression Adjustments for Estimating the Global Treatment Effect in Experiments with Interference. *Journal of Causal Inference* 7 (05 2019).

[6] David Roxbee Cox. 1958. *Planning of Experiments.* Wiley.

[7] Alex Deng, Ulf Knoblich, and Jiannan Lu. 2018. Applying the Delta Method in Metric Analytics: A Practical Guide with Novel Ideas. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (London, United Kingdom) *(KDD 2018)*. Association for Computing Machinery, New York, NY, USA, 233–242.

[8] Alex Deng, Ya Xu, Ron Kohavi, and Toby Walker. 2013. Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining* (Rome, Italy) *(WSDM '13)*. Association for Computing Machinery, New York, NY, USA, 123–132.

[9] Nikos Diamantopoulos, Jeffrey Wong, David Issa Mattos, Ilias Gerostathopoulos, Matthew Wardrop, Tobias Mao, and Colin McFarland. 2019. Engineering for a Science-Centric Experimentation Platform. arXiv:1910.03878 [cs.SE]

[10] Dean Eckles, Brian Karrer, and Johan Ugander. 2014. Design and Analysis of Experiments in Networks: Reducing Bias from Interference. *Journal of Causal Inference* (04 2014).

[11] Facebook. 2020. *Commuting Zones.* https://dataforgood.fb.com/tools/commutingzones/.

[12] Huan Gui, Ya Xu, Anmol Bhasin, and Jiawei Han. 2015. Network A/B Testing: From Sampling to Estimation. In *Proceedings of the 24th International Conference on World Wide Web* (Florence, Italy) *(WWW 2015)*. International WWW Conferences Steering Committee, Republic and Canton of Geneva, CHE, 399–409.

[13] Richard J Hayes and Lawrence H Moulton. 2017. *Cluster randomised trials.* CRC press.

[14] David Holtz, Ruben Lobel, Inessa Liskovich, and Sinan Aral. 2020. Reducing Interference Bias in Online Marketplace Pricing Experiments. arXiv:2004.12489 [stat.ME]

[15] Michael G Hudgens and M. Elizabeth Halloran. 2008. Toward Causal Inference With Interference. *J. Amer. Statist. Assoc.* 103, 482 (2008), 832–842.

[16] Igor Kabiljo, Brian Karrer, Mayank Pundir, Sergey Pupyrev, Alon Shalita, Alessandro Presta, and Yaroslav Akhremtsev. 2017. Social hash partitioner: a scalable distributed hypergraph partitioner. *arXiv preprint arXiv:1707.06665* (2017).

[17] Winston Lin. 2013. Agnostic notes on regression adjustments to experimental data: Reexamining Freedman's critique. *Ann. Appl. Stat.* 7, 1 (03 2013), 295–318.

[18] Joel A. Middleton and Peter M. Aronow. 2015. Unbiased Estimation of the Average Treatment Effect in Cluster-Randomized Experiments. *Statistics, Politics and Policy* 6, 1-2 (2015), 39 – 75.

[19] Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical Review E* 69, 2 (2004), 026113.

[20] J Pouget-Abadie, G Saint-Jacques, M Saveski, W Duan, S Ghosh, Y Xu, EM Airoldi, et al. 2019. Testing for arbitrary interference on experimentation platforms. *Biometrika* 106, 4 (2019), 929–940.

[21] David Rolnick, Kevin Aydin, Jean Pouget-Abadie, Shahab Kamali, Vahab Mirrokni, and Amir Najmi. 2019. Randomized Experimental Design via Geographic Clustering. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) *(KDD 2019)*. Association for Computing Machinery, New York, NY, USA, 2745–2753.

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

[23] Donald B. Rubin. 1980. Randomization Analysis of Experimental Data: The Fisher Randomization Test Comment. *J. Amer. Statist. Assoc.* 75, 371 (1980), 591–593.

[24] Martin Saveski, Jean Pouget-Abadie, Guillaume Saint-Jacques, Weitao Duan, Souvik Ghosh, Ya Xu, and Edoardo M. Airoldi. 2017. Detecting Network Effects: Randomizing Over Randomized Experiments. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD 2017)*. Association for Computing Machinery, New York, NY, USA, 1027–1035.

[25] Johan Ugander, Brian Karrer, Lars Backstrom, and Jon Kleinberg. 2013. Graph Cluster Randomization: Network Exposure to Multiple Universes. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Chicago, Illinois, USA) *(KDD 2013)*. Association for Computing Machinery, New York, NY, USA, 329–337.

[26] Huizhi Xie and Juliette Aurisset. 2016. Improving the Sensitivity of Online Controlled Experiments: Case Studies at Netflix. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 645âĂŞ654.

[27] Ya Xu, Nanyu Chen, Addrian Fernandez, Omar Sinno, and Anmol Bhasin. 2015. From Infrastructure to Culture: A/B Testing Challenges in Large Scale Social Networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia) *(KDD 2015)*. Association for Computing Machinery, New York, NY, USA, 2227–2236.

8 SUPPLEMENTAL MATERIAL

8.1 Implementation: Cluster management

Besides for the component that deploys treatments that is described in the main text, the other main component of the network experiment implementation is cluster management. We describe this more systems-oriented component here for completeness.

Clusterings can become stale or decrease in quality over time. New unclustered units can arrive to an online service, causing a divergence between the experiment's population and the online service's population. In addition, the pattern of interference can change over time, as might be detected by an increase in interactions between clusters. To account for this, the framework provides two options: create a new universe or refresh the clustering used by an existing universe.

To accomplish these options, we have pipelines that periodically regenerate clusterings. A particular set of clusters is indexed both by the name and date of creation. The clustering name denotes a sequence of clusterings generated over time that can be compared and contrasted. As an example, we might cluster the Facebook friendship graph on a weekly basis. The key-value service that provides clustering lookups is indexed by both the name and the date to identify the particular clusters.

The universe is assigned a clustering name and date upon creation. To refresh a universe by updating the date, all experiments need to be halted within that universe to avoid changing the treatment assignments of running experiments. We can then swap out the date of the clusters, while maintaining the clustering name. This refresh provides continuity to experimenters who understand a particular universe as being appropriate for their experiments.

8.2 Analysis: note on model-based alternatives

Agnostic design-based analysis is of course not the only way to analyze experiments. A class of alternatives are model-based approaches, which broadly speaking, fit a model to experimental results, and use the fitted model to extrapolate the global average treatment effect τ.

For example, regression modeling, like that described by [5], would attempt to estimate $\mu(\mathbf{w}, \mathbf{x}) = \mathbb{E}[Y_u|\mathbf{W} = \mathbf{w}, \mathbf{X} = \mathbf{x}]$ from the results of an experiment, by assuming that the dependence of μ on $\mathbf{w}$ and observed covariates $\mathbf{x}$ can be summarized through features, such as the number of nearby units in treatment to the random unit u, and covariates of u and their nearby units. If this model including covariates and features based on treatment conditions was correct (and was learned from data), evaluating it at $\mathbf{w} = \mathbf{1}$ and $\mathbf{w} = \mathbf{0}$ could be used to estimate the global average treatment effect τ. Unlike the agnostic regression adjustment, which we just used for improving precision, this approach requires trusting the model is well-specified enough to be useful.

A tremendous advantage of model-based analysis is that it produces an actual estimate of the global average treatment effect τ, as opposed to $\tau_{cluster}$. However, we are not aware of a satisfactory solution to fitting models, performing model validation and criticism across many experiments in many domains at scale. Considering that experts can have difficulty constructing and validating such causal models, asking engineers to do it themselves is a non-starter. Given this state, we view agnostic regression adjustment as a safe default that trades bias for robustness, scale, and simplicity. This may not be the best approach for any particular intervention, but is less likely to be dramatically wrong than extrapolation from an ill-formulated model. That being said, nothing in our framework precludes experts from building models on top of data generated by a network experiment. Modeling is an analysis choice that can be complementary to cluster-randomized designs.

8.3 Cluster design: cluster size distribution

Fig. 6 shows the cluster size distributions from Louvain (left, iteration = 3, similar patterns for other iterations) and from recursive BP (right). In general, Louvain follows a heavy-tailed distribution, while BP results in a much more balanced size distribution. We observe smaller resolution leads to smaller clusters in the tail but more clusters in the middle size range. On the left (small) side of the distribution, it follows very closely with the distribution from small connected components (CC) shown by the green dashed line, as expected from modularity maximization. The peak around 50 in the CC distribution is an artifact of filtering applied to the graph. From the right plot, BP maintains a tight distribution of cluster sizes across levels, with a shifting mean cluster size.

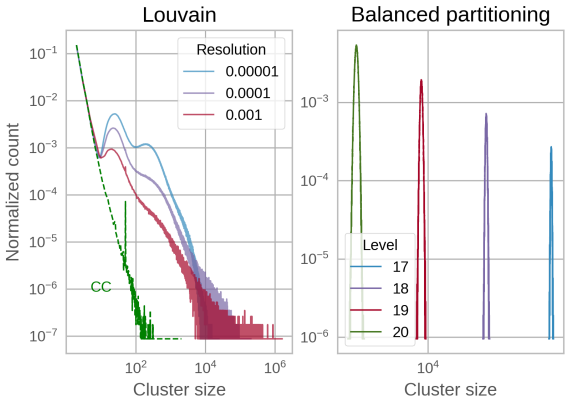

Figure 6: Cluster size distribution from (left) Louvain with different resolution and (right) recursive BP at different splitting level. Y-axis is the number of clusters at a specific cluster size, normalized by the total number of clusters.

8.4 Commuting Zones experiment: additional results

Since jobs on the platform are associated with a location, each job can also be assigned to a commuting zone. With the commuting zone experiment, the team could directly measure the effect of their change on job listings and employers. More specifically, we could test whether providing applications to jobs with few applications leads employers to post another job. To understand these effects, we focus on jobs that were posted before the test started, as well as employers who had posted a job before the test started. Including jobs that were created during the experiment would pose a problem if employers do in fact post more as a result of receiving more

applications. In this analysis, jobs and pages are considered to be exposed to the experiment as soon as the first user in their commuting zone is logged as triggered.

In the experiment, we observe that by increasing applications to jobs with few overall applications by 49.7%, the probability that a job receives an application increased by 14.069% ± 11.198% and the probability that an employer posted another job increased by 16.959% ± 15.731%. Both of these results were statistically significant at the 5% level, and were only visible due to the cluster-randomization.