NBER WORKING PAPER SERIES

USING MACHINE LEARNING FOR EFFICIENT FLEXIBLE REGRESSION ADJUSTMENT IN ECONOMIC EXPERIMENTS

John A. List
Ian Muir
Gregory K. Sun

Working Paper 30756
http://www.nber.org/papers/w30756

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
December 2022, Revised June 2024

We thank Lyft Inc. for providing a large portion of the data used in this project. We additionally thank Adeline Sutton for her help in accessing and interpreting the CHECC data as well as Brent Hickman, Michael Cuna, Atom Vayalinkal, and participants at the Advances in Field Experiments conference for helpful comments which have improved the paper. Documentation of our procedures and our Stata and R code can be found here: https://github.com/gsun593/ FlexibleRA The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

At least one co-author has disclosed additional relationships of potential relevance for this research. Further information is available online at http://www.nber.org/papers/w30756

Using Machine Learning for Efficient Flexible Regression Adjustment in Economic Experiments
John A. List, Ian Muir, and Gregory K. Sun
NBER Working Paper No. 30756
December 2022, Revised June 2024
JEL No. C9,C90,C91,C93

ABSTRACT

This study investigates the optimal use of covariates in reducing variance when analyzing experimental data. We show that finding the variance-minimizing strategy for making use of pre-treatment observables is equivalent to estimating the conditional expectation function of the outcome given all available pre-randomization observables. This is a pure prediction problem, which recent advances in machine learning (ML) are well-suited to tackling. Through a number of empirical examples, we show how ML-based regression adjustments can feasibly be implemented in practical settings. We compare our proposed estimator to other standard variance reduction techniques in the literature. Two important advantages of our ML-based regression adjustment estimator are that (i) they improve asymptotic efficiency relative to other alternatives, and (ii) they can be implemented automatically, with relatively little tuning from the researcher, which limits the scope for data-snooping.

John A. List
Department of Economics
University of Chicago
1126 East 59th
Chicago, IL 60637
and Australian National University
and also NBER
jlist@uchicago.edu

Ian Muir
Walmart
muir.ian.m@gmail.com

Gregory K. Sun
Washington University in St Louis
greg.s@wustl.edu

1 Introduction

Since the pioneering work of Fisher (1935), it has been well established in the scientific community that experimentation through randomized controlled trials is the gold standard through which it is possible to learn about cause and effect. However, the large subsequent literature on the nuances of experimental methodology illustrates that randomization is merely the beginning, not the end, of the experimental process. After leveraging advantages of controlling the assignment mechanism, many decisions remain: the population of people and situations to examine, how many units to include, what outcomes and observables to collect, and how to best examine and model the collected data, to name a few.

This paper focuses on the problem of maximizing statistical power, given an existing experimental design. Concretely, suppose the researcher has conducted an experiment aimed at affecting outcome Y through treatment $\mathbf{W}$ and that the researcher has collected data on covariates X which are independent of treatment status. The goal of this paper is twofold. Theoretically, we provide a transparent derivation of the best-practices in this setting and use our derivation to highlight what the sources of inefficiency are in common alternative variance reduction techniques. Our practical contribution is to show that machine-learning techniques, properly used, largely automate away the implementation of the best-practices identified by the theory.

Our theoretical results contribute to a large literature in statistics and econometrics that provides advice on how to optimally perform "covariate adjustment" or "regression adjustment" given the data setting described in the previous paragraph. Many papers in this literature study the asymptotic properties of estimators derived in specific special cases of the above framework. Frison and Pocock (1992); Tsiatis et al. (2008); Wager et al. (2016), and Negi and Wooldridge (2020), Roth and Sant'Anna (2023), and Cohen and Fogarty (2023) consider regression adjustments when the estimand is an average treatment effect. Negi and Wooldridge (2021) consider a more general case where one is potentially interested in estimating all potential outcome means separately. Their theoretical results center around comparing the asymptotic variance of three specific estimators, all of which are built off of linear regression. Hahn (1998) and Armstrong (2022) provide upper bounds on the maximal attainable variance reduction in these settings.

The most general asymptotic results that we are aware of within this literature are due to Zhang et al. (2008), who derive the variance minimizing regression adjustment within a semi-parametric framework. Their proposed estimator is similar to ours, and both attain the constrained asymptotic efficiency bounds derived in Armstrong (2022).[1] Our efficiency

[1] Specifically, our estimators attain the asymptotic efficiency bound subject to the constraint that

results can be thought of as a special case of theirs, but our derivations are structured so as to be as transparent as possible. Current research practice often diverges substantially from what is known to be optimal, and our approach helps clearly highlight *why* their current practices are sub-optimal. Our derivations allow us to characterize the sources of inefficiency within each of these practices.

We compare the optimal regression adjustment to five common practices in the literature: two-way-fixed effects, OLS using covariates without interactions, OLS with interactions, non-linear regression adjustment, and partially linear regression. In the body of the paper, we more formally define these commonly used alternatives and show how they relate to one another. Our results are summarized in Figure 1, which depicts the ordering of various estimators in the literature in terms of asymptotic variance and provides brief intuition for where efficiency loss from these practices stem from. As will be made more precise later, the optimal regression adjustment is related to estimating the conditional expectation function of outcome Y given covariates X, and the power loss of all commonly used variance reduction strategies come from implicitly using different approximations to those conditional expectation functions. When the approximations implicitly made by these alternative estimators are sufficiently "high quality" at approximating the conditional expectation function, the efficiency loss from not using the optimal RA are minimal. Our results clarify when this is likely to be the case.

Our paper also makes a number of contributions to the practical aspects of implementing regression adjustment. First, the optimal regression-adjusted estimators we derive take on a fairly straightforward form. Thus, the task of optimally making use of covariate information can be largely automated.[2]. Second, our approach generalizes a number of alternative practical implementations in the literature (Glynn and Quinn (2010); Rosenblum and Van Der Laan (2010); Pitkin et al. (2013); Wager et al. (2016); Poyarkov et al. (2016); Spiess (2018); Wu and Gagnon-Bartsch (2018); Rothe (2020); Opper (2021); Guo et al. (2021), and Jin and Ba (2023)). We do not review each of these prior estimators in detail, but note that all of the estimators in these prior papers are either asymptotically inefficient (either because they make parametric approximations to the conditional-expectation function, or because they pool observations across treatment groups) or they are constructed for estimating only

$\Pr(W_{i,g} = 1|X_i = x) = \rho_g$ for fixed proportions $\boldsymbol{\rho}$, and for all x. If randomization probabilities can be made conditional on x, then for a fixed target parameter, variance can be further decreased by exploiting heteroskedasticity in $Y_i(g)$ conditional on X_i. For instance, if the researcher is interested in estimating the average treatment effect, then the researcher could further reduce variance by over-sampling treatments for which the outcome of the variance is higher: $\Pr(W_{i,g} = 1|X_i = x) \propto \text{Var}(Y_i(g)|X_i = x)$. This information is often difficult to obtain in practice, and moreover, the optimal sampling design for one target parameter may not be optimal for another.

[2] We provide code for doing so at https://github.com/gsun593/FlexibleRA

particular parameters of interest, such as average treatment effects. Our estimator is efficient for any target parameter that can be written as a function of potential outcome means. This may be of particular interest to economists, who increasingly use experimental variation as a building block for estimating deeper structural primitives of the situations they are interested in (DellaVigna et al. (2012); Duflo et al. (2012); Cotton et al. (2020); Goldszmidt et al. (2021), and Bodoh-Creed et al. (2023), for instance). We conduct an extensive set of simulation studies and real-world data analysis exercises to highlight the finite-sample considerations that arise when implementing regression adjustment in practice. Our simulation results suggest two main takeaways for applied researchers. First, the confidence intervals constructed using our proposed methodology appear reliable and are typically smaller than those arising from using common alternatives. These results are achieved without explicit specification search on the part of the researcher, thus automating away much of the labor associated with reducing variance. Second, we show that when estimating average treatment effects, the percent reduction in variance is approximately equal to the R^2 of the ML model in predicting outcome Y given covariate X. This finding provides a useful heuristic for how a researcher who anticipates using our variance reduction methods can adjust their power calculations accordingly: the required sample size is smaller by a factor of $1 - R^2$ when using regression adjustment, relative to not using regression adjustment.

The plan for the rest of this paper is as follows. In Section 2, we develop our theory of regression adjustment. In Section 3, we use a number of empirical evaluations leveraging data from the ridesharing firm, Lyft, to show that the asymptotic theory provides a good guide for conducting inference and to show that the optimal choice of estimator can make a difference in practical settings. Section 4 concludes.

2 Theory

In this section, we develop the theory for our generalized regression adjustment. We begin with a standard potential outcomes model. The data are generated from an experiment with treatment groups $\{1, \ldots, G\}$. Let the potential outcome in group g be denoted by $Y(g)$ so that the objects of interest are the average potential outcome within each group, denoted

$$\mu_g = \mathbb{E}[Y(g)].$$

We assume that treatment is assigned via simple random sampling and define W_g to be an indicator that equals 1 if and only if an individual was randomized into group g and 0 otherwise. All individuals are assigned a treatment, so $\sum_{g=1}^{G} W_g = 1$. Let bold versions of letters be the vector obtained by stacking the versions with g subscripts so that, for example, $\mathbf{W} = (W_1, \ldots, W_G)'$ is the vector of treatments, and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_G)'$ is the vector of mean potential outcomes. We additionally assume that the researcher has access to a vector of pre-treatment covariates, X, which are potentially informative about the outcome Y. Our goal is to characterize the extent to which X can be used to reduce variance when estimating $\boldsymbol{\mu}$. Because we are working with an experiment, we assume the usual orthogonality condition arising from statistical independence:

Assumption 1.

$$\{Y(g)\}_{g=1}^{G}, X \perp \mathbf{W}.$$

We model the data as comprising an i.i.d. sample of size n from the experiment, denoted $\{Y_i, X_i, \mathbf{W}_i\}_{i=1}^{n}$, where $Y_i = Y_i(g_i)$ for the unique g_i satisfying $W_{i,g_i} = 1$. Finally, let $\rho_g = \Pr(W_g = 1)$ be the probability that an individual is in treatment group g. Therefore, $\boldsymbol{\rho}$ is the vector of treatment probabilities. Because we are discussing asymptotic variances, we maintain the following mild regularity condition throughout.

Assumption 2. $\mathbb{E}[Y(g)^2] < \infty$ for all $g = 1, \ldots, G$.

With the notation in hand, we can describe our theory in 5 sections. In Section 2.1, we describe a general class of regression-adjusted estimators and derive the optimal adjustment within this class. In Section 2.2, we use our derivations in 2.1 in order to relate the optimal regression adjustment to what is actually implemented in practice. In Section 2.3, we discuss how to feasibly implement the efficient estimator implied by Section 2.1 and derive the asymptotic distribution of this estimator, which we refer to as “flexible regression adjustment” (FRA). Section 2.4 discusses the choice of which ML method to use when implementing FRA. Finally, Section 2.5 provides practical guidance for how a researcher should do power calculations when they anticipate that they will use FRA when analyzing the data.

2.1 Efficiency Theory

This section derives the optimal regression adjustment for a broad class of estimators. To simplify exposition we begin our derivation by fixing one treatment group g, and focus on the task of estimating the mean potential outcome within that treatment group: $\mu_g = \mathbb{E}[Y(g)]$. Randomization and the potential outcomes model imply that $\mathbb{E}[Y(g)] = \mathbb{E}[Y|W_g = 1]$, so arguably the most straightforward estimator of μ_g is to take the sample analogue of the rightmost expression. This is the sample mean of the outcome in treatment group g, which we denote by $\bar{Y}_g \equiv \frac{1}{N_g}\sum_{i:W_{g,i}=1} Y_i$. While $\bar{Y}_g$ is unbiased and consistent, randomization of $\mathbf{W}$ with respect to *both* $Y(g)$ and with respect to X implies that it is not the only possible estimator of μ_g. Specifically, the fact that $\mathbf{W} \perp X$ implies that for any integrable function h of X, we have $\mathbb{E}[h(X)] = \mathbb{E}[h(X)|W_g = 1]$. This implies that we can alternatively write

$$\mu_g = \mathbb{E}[Y|W_g = 1] + \mathbb{E}[h(X)] - \mathbb{E}[h(X)|W_g = 1].$$

For any fixed function h_g (the g subscript emphasizes that h_g may be chosen separately for different treatment groups), denote the sample analogue of the above expression by $\hat{\mu}_g^{h_g}$, i.e.,

$$\hat{\mu}_g^{h_g} \equiv \bar{Y}_g + \frac{1}{N}\sum_i h_g(X_i) - \frac{1}{N_g}\sum_{i:W_{i,g}=1} h_g(X_i). \tag{1}$$

Estimators of the form (1) have the interpretation of "adjusting" the baseline mean estimate $\bar{Y}_g$ by subtracting off the degree to which the mean value of $h_g(X_i)$ differs in experiment group g compared to the overall experimental sample. In what follows, we will sometimes refer to h_g in the above expression as the *adjustment function.* This class of estimators for $\hat{\mu}_g^{h_g}$ does not cover every conceivable unbiased estimator for μ_g, but all estimators which are commonly used in practice are asymptotically equivalent to an estimator of the form (1). In the body of the paper, we will explicitly derive the most efficient estimator of the form in Equation (1), although in Appendix A.1, we confirm that our derived optimal estimator is also semi-parametrically efficient in the broader sense of Zhang et al. (2008).

An advantage of restricting ourselves to estimators of the form (1) is that the asymptotic variance of this class of estimators is highly tractable. To help simplify notation in what follows, for any arbitrary function f_g of X_i, we introduce the following notational convention:

$$\bar{f}_{g,all} = \frac{1}{n}\sum_{i=1}^{n} f_g(X_i), \quad \bar{f}_{g,g} = \frac{1}{n_g}\sum_{i:W_{i,g}=1} f_g(X_i).$$

In the above notation, the first subscript is the same subscript as in f_g, while the second subscript indicates the subsample for which the sample mean is taken. Given this notation, we may write $\hat{\mu}_g^{h_g} = \bar{Y}_g + \bar{h}_{g,all} - \bar{h}_{g,g}$.

We now give an explicit expression for the asymptotic variance of estimators of the form $\hat{\mu}_g^{h_g}$. To do so, we first decompose $Y_i(g)$ as equaling its conditional expectation given X_i plus its deviations from that conditional expectation. Let

$$m_g(x) \equiv \mathbb{E}[Y|X = x, W_g = 1], \quad \text{so that} \quad Y_i(g) = m_g(X_i) + \varepsilon_i(g),\ \mathbb{E}[\varepsilon_i(g)|X_i] = 0. \tag{2}$$

Then denoting $d_g(x) = h_g(x) - m_g(x)$, we can rearrange $\hat{\mu}_g^{h_g}$ as

$$\hat{\mu}_g^{h_g} = \bar{\varepsilon}_g + \bar{m}_{g,g} + \bar{h}_{g,all} - \bar{h}_{g,g} = \underbrace{\bar{\varepsilon}_g}_{A_g} + \underbrace{\bar{m}_{g,all}}_{B_g} + \underbrace{(\bar{d}_{g,all} - \bar{d}_{g,g})}_{C_g^{h_g}}, \tag{3}$$

where $\bar{\varepsilon}_g = \frac{1}{n_g} \sum_{i:W_{i,g}=1} \varepsilon_i(g)$ is the mean value of $\varepsilon_i(g)$ among individuals in treatment g. Since $\varepsilon_i(g)$ is mean independent of X_i, A_g is uncorrelated with $B_{g'}$ and $C_{g'}^{h_g}$ for any g, g' and for all choices of d_g. To see why, note that mean independence implies that $\mathrm{Cov}(\varepsilon_i(g), f(X_i)) = 0$ for any function f of X_i and for any $i \neq j$, our i.i.d. sampling assumption implies that $\mathrm{Cov}(\varepsilon_i(g), f(X_j)) = 0$. But because the summands comprising B_g and $C_g^{\boldsymbol{h}}$ are all functions of X_i, this shows that all summands comprising A_g are uncorrelated with all summands comprising B_g and $C_g^{\boldsymbol{h}}$.

Additionally, B_g is uncorrelated with $C_{g'}^{\boldsymbol{h}}$, again for all g, g' and for any fixed choice of $\mathbf{h}$. To see why, note that because $\mathbb{E}[\bar{d}_{g',all} - \bar{d}_{g',g'}] = 0$,

$$\begin{aligned}
\mathrm{Cov}\left(B_g, C_{g'}^{\boldsymbol{h}}\right) &= \mathbb{E}[\bar{m}_{g,all}(\bar{d}_{g',all} - \bar{d}_{g',g'})] \\
&= \frac{1}{n^2}\left(\underbrace{\sum_{i=1}^{n} \mathbb{E}[m_g(X_i)d_{g'}(X_i)]}_{S_1} + \underbrace{\sum_{i \neq j} \mathbb{E}[m_g(X_j)d_{g'}(X_i)]}_{S_2}\right) \\
&\quad - \frac{1}{nn_{g'}}\left(\underbrace{\sum_{i:W_{i,g'}=1} \mathbb{E}[m_g(X_i)d_{g'}(X_i)]}_{S_3} + \underbrace{\sum_{i:W_{i,g'}=1} \sum_{j \neq i} \mathbb{E}[m_g(X_j)d_{g'}(X_i)]}_{S_4}\right).
\end{aligned} \tag{4}$$

As is clear, the summands in S_1 and S_3 are identical, as are the summands of S_2 and S_4. Moreover, S_1 has n elements, S_2 has $n(n-1)$ elements, S_3 has $n_{g'}$ elements, and S_4 has $n_{g'}(n-1)$ elements. It is then straightforward to see that everything in the above expression cancels, so the covariance vanishes, as desired.

This reveals that the three terms in (3) are all uncorrelated with each other. Stacking Equation (3) for all of the g into a single equation and using the fact that the variance of the sum of uncorrelated random vectors is the sum of the variances, we obtain the following variance decomposition

$$\mathrm{Var}\left(\hat{\boldsymbol{\mu}}^{\mathbf{h}}\right) = \mathrm{Var}\left(\mathbf{A}\right) + \mathrm{Var}\left(\mathbf{B}\right) + \mathrm{Var}\left(\mathbf{C}^{\mathbf{h}}\right). \tag{5}$$

Note that $\mathbf{h}$ does not affect $\mathbf{A}$ or $\mathbf{B}$ and therefore only affects the asymptotic variance of the regression adjusted estimator by affecting $\mathbf{C}^{\mathbf{h}}$. Moreover, it must contribute a positive semi-definite matrix, which is minimized if we can choose a value of $\mathbf{h}$ making $\text{Var}(\mathbf{C}^{\mathbf{h}}) = 0$. This is exactly what happens when we set $h_g = m_g$ for all g.[3] We have thus shown that within the class of regression adjustment estimators of the form (1), $\hat{\boldsymbol{\mu}}^{\mathbf{m}}$ results in the lowest variance, where $\mathbf{m} = (m_1, \ldots, m_G)'$ is the vector of conditional expectation functions. In Appendix A.1, we prove the following result, showing that $\hat{\boldsymbol{\mu}}^{\mathbf{m}}$ is efficient in an even broader sense formalized by the following result:

Proposition 1. *Let $\tilde{\boldsymbol{\mu}}$ be an estimator of $\boldsymbol{\mu}$ which is consistent and asymptotically normal under any data-generating process satisfying Assumptions 1 and 2. Then* $\text{Var}(\tilde{\boldsymbol{\mu}}) \geq \text{Var}(\hat{\boldsymbol{\mu}}^{\mathbf{m}})$.

Remark 1. The argument leading to the variance decomposition (5) continues to hold if Y is vector valued, so in that case, the variance minimizing choice of $\mathbf{h}$ continues to be the conditional expectation of each component of Y for each group g.

In practice, $\mathbf{m}$ is unknown, and thus must be estimated. In Section 2.3, we show that the asymptotic variance of an estimator formed by plugging in a fitted value $\hat{\mathbf{m}}$ of $\mathbf{m}$ into $\hat{\boldsymbol{\mu}}^{\mathbf{m}}$ yields an estimator with the same asymptotic variance. Before discussing inference in more detail, however, in the next subsection, we compare the efficient regression adjustment derived in this subsection with common alternatives used in practice.

2.2 Inefficiency of Commonly Used Treatment Effect Estimators

In the previous section, we derived the theoretically optimal regression adjustment. In this section, we use the variance decomposition from the previous section to shed further light on the most common variance reduction strategies used by practitioners. Our framework allows us to analytically characterize the sources of inefficiency in these approaches. All of these common variance reduction strategies are (asymptotically) of the form of Equation (1), but use different approximations to the CEF for the adjustment function h_g. These alternative estimators are often simpler to use than ours, and their implicit approximations may be reasonable in some settings, thus potentially justifying their usage, despite asymptotic inefficiency. Our discussion here sheds light on when this explanation is plausible.

In order to make our analysis more tractable, we focus on the simplest, but most common case: a binary treatment where the parameter of interest is the average treatment effect. Thus, assume that $W \in \{0, 1\}$, and the parameter of interest is $\mu_1 - \mu_0$. Let $\rho_0 = \Pr(W = 0)$

[3] Such a choice makes $\mathbf{C}^{\mathbf{h}}$ deterministically 0.

and $\rho_1 = \Pr(W = 1) = 1 - \rho_0$. As we will see, all of the most common estimators in the literature take the form of $\hat{\mu}_1^{h_1} - \hat{\mu}_0^{h_0}$ for some adjustment function h_1, h_0. Using decomposition (3), we can thus write

$$\hat{\mu}_1^{h_1} - \hat{\mu}_0^{h_0} = (\bar{\varepsilon}_1 - \bar{\varepsilon}_0) + (\bar{m}_{1,all} - \bar{m}_{0,all}) + [(\bar{d}_{1,all} - \bar{d}_{1,1}) - (\bar{d}_{0,all} - \bar{d}_{0,0})].$$

The variance of this expression only depends on the variance of the last term. Recall that the efficient regression adjustment sets this last term to 0. Thus, for any other treatment effect estimator, the degree of inefficiency can be characterized in terms of the size of this last term, which, after some algebraic manipulations, can be simplified as follows:

Lemma 1. *Let $h_1(x), h_0(x)$ be two measurable functions of x. Then as $N \to \infty$,*

$$N[\mathrm{Var}(\hat{\mu}_1^{h_1} - \hat{\mu}_0^{h_0}) - \mathrm{Var}(\hat{\mu}_1^{m_1} - \hat{\mu}_0^{m_0})] \to \frac{\mathrm{Var}(\rho_0 d_1(X_i) + \rho_1 d_0(X_i))}{\rho_0 \rho_1}. \tag{6}$$

Proof. See Appendix A.2. □

The expression inside the variance operator in (6) is a weighted average of the approximation error of h_1 and h_0 in representing the conditional expectation functions of Y given X in treatment and control. Lemma 1 shows that we can always think of the degree of inefficiency as being proportional to the variance of this weighted error.

We now analyze the sources of inefficiency from a number of common regression adjustment techniques. We begin with the partially linear model (PLM), where researchers (see, e.g., Urminsky et al. (2016)) estimate a regression of the form

$$Y = \beta W + g(X) + \varepsilon.$$

using double machine learning techniques. The resulting estimator is asymptotically equivalent to the following infeasible estimator:

$$\frac{1}{N}\sum_{i=1}^{N} \frac{(Y_i - \mathbb{E}[Y|X_i])(W_i - \rho_1)}{(W_i - \rho_1)^2}.$$

By the law of iterated expectations, $\mathbb{E}[Y|X_i] = \rho_0 m_0(X_i) + \rho_1 m_1(X_i)$. Rearranging the above expression shows that this estimator is asymptotically equivalent to an estimator of the form (1), with adjustment functions given by $h_1(x) = h_0(x) = \rho_0 m_0(x) + \rho_1 m_1(x)$. Thus, we have $d_0(x) = \rho_1(m_1(x) - m_0(x))$ while $d_1(x) = \rho_0(m_0(x) - m_1(x))$. Intuitively, the PLM differs from the fully efficient regression adjustment by using a pooled adjustment function $\mathbb{E}[Y|X]$, rather than adjusting separately for each treatment-group specific CEF.

Using Equation (6), the degree of inefficiency arising from partially linear regression is

$$\frac{(\rho_1^2 - \rho_0^2)^2}{\rho_0 \rho_1} \mathrm{Var}(m_1(X_i) - m_0(X_i)). \tag{7}$$

The first factor $\frac{(\rho_1^2 - \rho_0^2)^2}{\rho_0 \rho_1}$ measures the degree to which the sizes of treatment and control groups are imbalanced while $\mathrm{Var}(m_1(X_i) - m_0(X_i))$ measures the extent of treatment effect heterogeneity explainable by X_i. If either *i*) conditional average treatment effects are constant, or *ii*) the probability of being in treatment is exactly a $\rho_1 = 0.5$, then PLM is asymptotically equivalent to the efficient regression adjustment. Otherwise, due to the fact that it adjusts by a pooled conditional expectation function may lead to inefficiencies.

We next turn our attention towards a number of common linear regression adjustments. We begin with the "separate regression adjustment" of Negi and Wooldridge (2020) (henceforth NW). This regression adjustment is defined as follows. First, regress Y on X in the control group ($W = 0$), and call the resulting coefficients from this regression $\hat{\alpha}_0, \hat{\beta}_0$. Second, regression Y on X in the treatment group ($W = 1$), and call the resulting coefficients from this regression $\hat{\alpha}_1$,$\hat{\beta}_1$. Finally, estimate the treatment effect as $(\hat{\alpha}_1 - \hat{\alpha}_0) + \bar{X}(\hat{\beta}_1 - \hat{\beta}_0)$. Alternatively, we can write the separate regression adjustment as

$$(\hat{\alpha}_1 + \hat{\beta}_1' \bar{X}_1) + \hat{\beta}_1'(\bar{X} - \bar{X}_1) - ((\hat{\alpha}_0 + \hat{\beta}_0' \bar{X}_0) + \hat{\beta}_0'(\bar{X} - \bar{X}_0)).$$

Properties of OLS imply that $\bar{Y}_g = \hat{\alpha}_g + \hat{\beta}_g' \bar{X}_g$, so the above estimator can be written as

$$\bar{Y}_1 + \hat{\beta}_1'(\bar{X} - \bar{X}_1) - (\bar{Y}_0 + \hat{\beta}_0'(\bar{X} - \bar{X}_0)).$$

This estimator again takes the same form as (1), with $h_0(x) = \hat{\alpha}_0 + \hat{\beta}_0' x$ and $h_1(x) = \hat{\alpha}_1 + \hat{\beta}_1' x$.

This representation, combined with Equation (6) shows that the degree of inefficiency from NW's separate regression adjustment is proportional to the variance of a weighted average of approximation errors in the CEFs of Y given X in treatment and control. Particularly illuminating is the case where there is no treatment effect heterogeneity, so that $m_1(X) = m_0(X) + \alpha$. In this case, $d_1(X) = d_0(X)$, and hence, the degree of inefficiency from the NW procedure is simply proportional to the mean squared approximation error from using a linear approximation to the CEF.

Despite these efficiency losses, NW's separate regression adjustment minimizes variance within an appropriately defined class of *linear* regression adjustments. Specifically, we say that a regression adjustment is *linear* if it takes the form of (1), but the adjustment functions h_0 and h_1 are linear in x. Abusing notation slightly, if β_g is a vector with the same dimensionality as x, define $\hat{\mu}_g^{\beta_g}$ as the regression adjustment where $h_g(x) = \beta_g' x$. Then $\hat{\mu}_g^{\beta_g} = \bar{Y}_g + \beta_g'(\bar{X} - \bar{X}_g)$. We generalize NW's results by showing that their separate regression adjustment is optimal relative to the class of linear regression adjustments. To do so, we define the *best linear predictor* of Y given X in treatment group g as

$$\text{BLP}_g(Y|x) = \alpha_g^* + (\beta_g^*)'x, \quad \alpha^*, \beta^* = \operatorname*{argmin}_{\alpha,\beta} \mathbb{E}\left[(Y(g) - \alpha - \beta' X)^2\right].$$

Define the error from the best linear predictor as $\delta_i(g) \equiv Y_i(g) - \text{BLP}_g(Y|X_i)$. This error is not mean independent of X like $\varepsilon_i(g)$ is, but it is uncorrelated. Let $\ell_g(x) \equiv \text{BLP}_g(Y|x)$. Then the following decomposition is the analogue of (3) for the linear case:

$$\hat{\mu}_g^{\beta_g} = \bar{\delta}_g + \bar{\ell}_{g,g} + \bar{h}_{g,all} - \bar{h}_{g,g} \equiv \underbrace{\bar{\delta}_g}_{A_g} + \underbrace{\bar{\ell}_{g,all}}_{B_g} + \underbrace{(\bar{d}_{g,all} - \bar{d}_{g,g})}_{C_g^{\beta_g}}, \tag{8}$$

where now, we define $d_g(x) = \beta_g' x - \ell_g(x)$. It is still the case that A_g is uncorrelated with $B_{g'}$ and $C_{g'}^{\beta_{g'}}$ for any choice of g, g', β_g, and $\beta_{g'}$.[4] Similarly, the argument for why B_g is uncorrelated with $C_g^{\beta_g}$ remains unchanged as well. Again, stacking the equations for all of the g's together, we arrive at the following analogue of Equation (5)

$$\text{Var}\left(\hat{\boldsymbol{\mu}}^{\boldsymbol{\beta}}\right) = \text{Var}\left(\mathbf{A}\right) + \text{Var}\left(\mathbf{B}\right) + \text{Var}\left(\mathbf{C}^{\boldsymbol{\beta}}\right). \tag{9}$$

The choice of regression coefficient $\boldsymbol{\beta}$ only affects the third term, so the variance of the above expression can be minimized if the third term can be set to 0. This is what happens when β_g is the OLS slope within group g, showing that NW's separate regression adjustment has lower variance than any other linear regression adjustment.

Analogous to our analysis of the PLM above, we next analyze what NW call the "pooled regression adjustment", where researchers estimate treatment effects as the coefficient from the following linear regression:

$$Y = \beta W + \boldsymbol{\gamma}' X + \varepsilon.$$

[4] Note a subtle difference in the justification for this fact. In this case, A_g is uncorrelated only with linear functions of X, but because we are restricting ourselves to the class of linear in X regression adjustments, the summands of B_g and $C_g^{\beta_g}$ are all restricted to be linear as well.

This resulting regression adjustment is asymptotically equivalent to

$$\frac{1}{N}\sum_{i=1}^{N}\frac{(Y_i-\ell(X_i))(W_i-\rho_1)}{(W_i-\rho_1)^2}.$$

where $\ell(x) = \text{BLP}(Y|x)$ is the best linear predictor of Y, given x, but when data from treatment and control are pooled together. It is thus equivalent to a linear regression adjustment where $\beta_1 = \beta_0 = \rho_0\beta_0^* + \rho_1\beta_1^*$ where β_0^*, β_1^* are respectively the population OLS slopes in control and treatment. We thus see that the pooled regression adjustment entails two sources of inefficiency. First, there is an efficiency loss from approximating the CEF as linear. Second, there is a further efficiency loss from using a pooled adjustment function instead of using group-specific adjustments. The efficiency loss from this second source is given by the following analogue of Equation (7)

$$\frac{(\rho_1^2-\rho_0^2)^2}{\rho_0\rho_1}\text{Var}(\ell_1(X_i)-\ell_0(X_i)).$$

We have thus re-derived one of the key results in NW's Theorem 5.2, which shows that pooled regression adjustment is less efficient than separate regression adjustment when $\rho_1 \neq 0.5$ and heterogeneity in treatment effects is predictable as a linear function of X_i. This intuition exactly mirrors the intuition behind when PLM is inefficient relative to the fully optimal regression adjustment.

We next analyze the use of panel regressions to reduce variance. Many experimental settings are naturally represented as having a panel structure. An outcome Y, is measured repeatedly across many time periods, and the experimenter intervenes for treatment units at some time period (e.g., Todd and Wolpin (2006), Kaplan et al. (2013), Fowlie et al. (2020), Gosnell et al. (2020)). For simplicity, we assume that there are $T_{pre}+1$ periods of time prior to intervention, and between periods $t = 0$ and $t = 1$, the experimenter randomly assigns some units to treatment and some units to control and observes all units for an additional T_{post} periods. The data from these settings can be represented as $\{(Y_{i,-T_{pre}},\ldots,Y_{i,-1},Y_{i,0},Y_{i,1},\ldots,Y_{i,T_{post}},W_i)\}_{i=1}^N$. Randomization implies that $(Y_{-T_{pre}},\ldots,Y_0) \perp W_i$, and hence, these pre-treatment observations satisfy the same statistical assumptions as what we have been refering to as "covariates" X. A common practice in these settings is to treat the data as a panel and to estimate treatment effects via a two-way fixed effects (2WFE) regression of the form:

$$Y_{i,t} = \alpha + \beta W_i \mathbb{1}\{t>0\} + \gamma_i + \delta_t + \varepsilon.$$

A standard decomposition of this estimator (e.g., see Goodman-Bacon (2021)) shows that the resulting estimate of the treatment effect takes the form of a difference-in-difference

$$(\bar{Y}_{1,post} - \bar{Y}_{0,post}) - (\bar{Y}_{1,pre} - \bar{Y}_{0,pre}),$$

where $\bar{Y}_{g,post}$ is the average value of Y in the post-treatment period ($t > 0$) in treatment group g while $\bar{Y}_{g,pre}$ is the average value of Y in the pre-tretment period ($t \leq 0$) in treatment group g. We can rearrange this estimator to be

$$(\bar{Y}_{1,post} + (\bar{Y}_{pre} - \bar{Y}_{1,pre})) - (\bar{Y}_{0,post} + (\bar{Y}_{pre} - \bar{Y}_{0,pre})),$$

where $\bar{Y}_{pre}, \bar{Y}_{post}$ are the unconditional mean outcomes in the pre/post periods respectively.

Denote individual i's average outcome before and after treatment respectively by $Y_{i,pre}$ and $Y_{i,post}$. The 2WFE estimator is thus a linear regression adjustment, where the outcome of interest is $Y_{i,post}$. The coefficients of the adjustment function are given by $\beta_0 = \beta_1 = \frac{1}{T_{pre}+1}$. The sources of inefficiency are three-fold. First, the form of the regression adjustment is linear, so again, even using the best possible linear regression adjustment entails an efficiency loss from taking a linear approximation to the CEF. Second, instead of using information about the entire trajectory of Y prior to treatment to predict $Y_{i,post}$, it only uses the average value, $Y_{i,pre}$. By an analogue of Equation (6), this entails a further efficiency loss of at least $\frac{\mathrm{Var}(\rho_0\epsilon_1+\rho_0\epsilon_0)}{\rho_0\rho_1}$, where $\epsilon_g = \mathrm{BLP}_g(Y_{i,post}|Y_{i,-T_{pre}}, \ldots, Y_{i,0}) - \mathrm{BLP}_g(Y_{i,post}|Y_{i,pre})$ for $g = 0, 1$. This second source of efficiency loss has an intuitive interpretation when treatment effects are constant: it is proportional to the change in R^2 when comparing a regression predicting Y using only $Y_{i,pre}$ as a single covariate compared to a regression using all pre-treatment observations as covariates. Third, even subject to only using $Y_{i,pre}$ as the sole predictor, the 2WFE regression is inefficient. The optimal linear regression that conditions only on $Y_{i,pre}$ should set the coefficients on the adjustment equal to the OLS slope from regression of $Y_{i,post}$ on $Y_{i,pre}$. Call the OLS slope in control β_0 and the OLS slope in treatment β_1. The 2WFE estimator, by contrast, sets this slope equal to 1. Again, appealing to an analogue of Equation (6), this final consideration leads to a further efficiency loss of

$$\frac{(\rho_0(\beta_1 - 1) + \rho_1(\beta_0 - 1))^2 \mathrm{Var}(Y_{i,pre})}{\rho_0\rho_1},$$

which is proportional to the square of a weighted average deviation of β_1 and β_0 from 1.

We consider one final set of variance reduction techniques, namely, nonlinear regression adjustment as suggested by NW. Because many of the ideas overlap with our discussion of linear regression adjustments, we focus solely on their full nonlinear regression adjustment.

The family of nonlinear (parametric) regression adjustments can be described as follows. First, in each treatment group, $g = 0, 1$, fit a nonlinear predictor of Y given X of the form $\hat{Y} = f(\hat{\alpha}_g + \hat{\beta}_g' X)$.[5] Second, take the regression-adjusted estimator of potential outcome mean in group g to be $\hat{\mu}_g^{fna} = \frac{1}{N}\sum_{i=1}^{N} f(\hat{\alpha}_g + \hat{\beta}_g' X_i)$. Throughout, the authors maintain the assumption that the estimating equations for $\hat{\alpha}_g, \hat{\beta}_g$ are such that the average fitted value of the regression function within group g equals the average outcome in group g, that is:

$$\bar{Y}_g = \frac{1}{N_g} \sum_{i:W_{i,g}=1} f(\hat{\alpha}_g + \hat{\beta}_g' \bar{X}_i).$$

This is a property possessed by OLS, but it also is similarly shared by quasi maximum-likelihood estimators in generalized linear models with a canonical link function. Using the same decomposition that we used for the separate nonlinear regression adjustment, we find

$$\hat{\mu}_g^{fna} = \bar{Y}_g + \frac{1}{N}\sum_{i=1}^{N} f(\alpha_g^* + (\beta_g^*)' X_i) - \frac{1}{N_g} \sum_{i:W_{i,g}=1} f(\alpha_g^* + (\beta_g^*)' X_i).$$

Nonlinear regression adjustment is thus asymptotically of the form of Equation (1), with $h_g(x) = f(\alpha_g^* + (\beta_g^*)' x)$, and where α_g^* and β_g^* are the population analogues of the regression coefficients within group g. This shows that the intuition for the efficiency loss from nonlinear regression adjustment is almost identical to the intuition for the efficiency loss from linear regression adjustment: in both cases, one is trying to approximate the CEF with a parametric form, but in the nonlinear case, the choice of parametric form is typically carefully picked to encode prior knowledge about Y (e.g., when Y is binary, logistic regression, rather than linear regression, may be better suited to modeling the CEF). To the extent that the choice of parameterization used in nonlinear regression adjustment is better adapted to the data-generating process of Y, we expect that it provides a better approximation to the CEF than OLS. Equation (6) in turn shows that in this case, we expect that nonlinear regression adjustment leads to greater variance reduction than linear regression adjustment. Indeed, NW's empirical findings seem to support this intuition.

Summarizing the discussion from this section, the variance reduction strategies commonly used in practice often differ from the optimal regression in a number of ways, and these estimators can be partially ranked. The partially linear regression maintains the flexibility of using ML methods, but pools together data from multiple treatments into a single adjustment function. The OLS with interactions estimator (separate regression adjustment) of NW uses

[5] The two examples NW explicitly have in mind are logistic regression and Poisson regression. In the former case, $f(\cdot) = \exp(\cdot)/(1 + \exp(\cdot))$ while in the latter case, $f(\cdot) = \exp(\cdot)$.

separate adjustment functions, but loses efficiency due to approximating the CEF as linear in X. The OLS without interactions estimator (pooled regression adjustment) further loses efficeincy due to using a pooled adjustment function instead of separate adjustments. The efficiency loss from the panel regression (2WFE) estimator loses efficiency from a number of sources; it only uses information about the average pre-treatment outcome, and it mechanically sets the coefficient on this average to 1. Finally, nonlinear regression adjustments lose efficiency because it approximates the CEF in such a way that a transformation is linear in X. The reader may refer again to Figure 1 for a visual summary of these relationships.

2.3 Sample Splitting and Inference

In this subsection, we discuss how to estimate the optimal regression adjustment derived in Section 2.1. In particular, we show that it is possible to tractably conduct inference while using a flexible non-parametric model for fitting the conditional expectation functions $m_g(X)$, including by making use of machine learning methods, which have demonstrated a remarkable ability to solve prediction problems in real world datasets.

A potential pitfall of estimating $m_g(X)$ too flexibly, which we deal with here, is that this flexibility can introduce substantial finite-sample bias in the estimates for $\boldsymbol{\mu}$ through a form of over-fitting bias. Fortunately, recent advances in semi-parametric estimation as summarized, for instance, in Chernozhukov et al. (2018) (henceforth CCDDHNR) provides exactly an approach to avoid this issue. Specifically, the results from CCDDHNR show that much of this finite sample bias may be avoided by making use of a sample-splitting scheme, which the authors call "k-fold cross fitting". The estimators we construct will closely follow the construction of CCDDHNR, adapted to the specifics of our settings.

To apply the insights in CCDDHNR to our setting, we begin by remarking on a useful property enjoyed by the class of regression adjustment estimators of the form (1): any estimator within this class of estimators has the same expected value. That is, for any two adjustment functions $\mathbf{h}, \mathbf{h}'$, $\mathbb{E}[\hat{\boldsymbol{\mu}}^{\mathbf{h}}] = \mathbb{E}[\hat{\boldsymbol{\mu}}^{\mathbf{h}'}]$. In particular, this implies that all second order Gateaux derivatives of $\hat{\boldsymbol{\mu}}^{\mathbf{h}}$ with respect to $\mathbf{h}$ vanish. In particular,

$$\left.\frac{\partial^2}{\partial r^2}\right|_{r=0} \mathbb{E}\left[\hat{\boldsymbol{\mu}}^{\mathbf{m}+r\mathbf{h}}\right] = 0, \ \forall \mathbf{h} \in L^2(X).$$

Then, appealing to Theorems 3.1 and 3.2 of CCDDHNR provides shows that

Proposition 2. *Suppose that* $\mathbb{E}[|Y|^q] < \infty$ *for some* $q > 2$. *Assume we have some procedure for generating* $\hat{\mathbf{m}}$ *which is consistent in the sense that* $||\hat{\mathbf{m}} - \mathbf{m}||_2 \xrightarrow{p} 0$. *Suppose that we estimate* $\hat{\mathbf{m}}$ *using a cross fitting procedure as described in CCDDHNR and plug in the fitted* *values of* $\hat{\mathbf{m}}$ *into our estimates* $\hat{\boldsymbol{\mu}}^{\mathbf{m}}$*. Then we have*

$$\sqrt{n}\left(\hat{\boldsymbol{\mu}}^{\hat{\mathbf{m}}} - \boldsymbol{\mu}\right) \xrightarrow{d} \mathcal{N}(0, V), \quad V = \mathrm{Var}(\mathbf{A}) + \mathrm{Var}(\mathbf{B}).^{6} \tag{10}$$

Moreover, V can be consistently estimated using sample variances and covariances as described below.

It is worth emphasizing that the conditions imposed in Proposition 2 are fairly weak. In particular, because of the higher order orthogonality of the class of regression adjustments to $\mathbf{m}$, we merely require that $\hat{\mathbf{m}}$ is consistent and do not require any conditions on how quickly the convergence occurs.[7] Many non-parametric estimators are known to be strongly consistent for fairly general classes of $\mathbf{m}$ (see, for instance, Györfi et al. (2002)). We, therefore, find that asymptotically, not much is lost by switching from using linear regression adjustment to a more flexible non-parametric regression adjustment with appropriate sample splitting.[8]

For completeness, we provide an explicit algorithm using Proposition 2 to estimate efficiently a regression adjustment and to obtain correct asymptotic standard errors. We use the sample splitting procedure suggested by CCDDHNR and randomly split the data into K roughly equal size folds.[9] We then fit $\hat{\mathbf{m}}$ in the following way:

1. For each fold $k \in 1, \ldots, K$, and for each g, fit $\hat{m}_g^{(-k)}(X)$ using a non-parametric method for estimating a conditional expectation function on data not in fold k, but in treatment group g.

2. For each index i in fold k, let $\hat{m}_{g,i} = \hat{m}_g^{(-k)}(X_i)$

Finally, we form the point estimate

$$\begin{aligned} \hat{\mu}_g^{FRA} &= \frac{1}{n_g} \sum_{i:W_{i,g}=1} (Y_i - \hat{m}_{g,i}) + \frac{1}{n} \sum_{i=1}^{n} \hat{m}_{g,i} \\ &= \frac{1}{n_g} \sum_{i:W_{i,g}=1} \underbrace{(Y_i - m_{g,i})}_{a_{g,i}} + \frac{1}{n} \sum_{i=1}^{n} \underbrace{m_{g,i}}_{b_{g,i}} + o(1/\sqrt{n}), \end{aligned} \tag{11}$$

[6] Where here, **A** and **B** are as in the previous section.

[7] As we will see in our simulations, we still prefer $\hat{\mathbf{m}}$ to be high quality, as the ability of $\hat{\mathbf{m}}$ to fit the data affects the sampling variability of the resulting estimator.

[8] However, this point should not be overstated. Nonparametric estimators typically suffer from slower rates of convergence than parametric estimators, so in a finite sample, one may still prefer linear regression adjustment. Our empirical results suggest that, in general, one should pick the method that produces the highest quality out-of-sample predictions of the outcome as measured by mean squared error.

[9] Note that if the sample size is not sufficiently large, some care should be taken to ensure that each fold gets observations from each of the treatment groups g.

where n_g is the number of observations in group g. We have now written each individual treatment group mean estimator in terms of two sample averages: one in the treatment sample and one in the full sample. Additionally, from the efficiency proof, we have $\text{Cov}(a_{g,i}, b_{g',j}) = 0$ for both $i = j$ and $i \neq j$. Computing the full covariance matrix between all of the groups is now simply an accounting exercise, where only covariances between terms from the same group need to be accounted for in the computation. Separately doing this for diagonal ($V_{g,g}$) and off diagonal ($V_{g,g'}$) terms gives us:

$$\begin{aligned} \hat{V}_{g,g} &= \frac{1}{n_g}\widehat{\text{Var}}(a_g) + \frac{1}{n}\widehat{\text{Var}}(b_g) \\ \hat{V}_{g,g'} &= \frac{1}{n}\widehat{\text{Cov}}(b_g, b_{g'}), \end{aligned} \tag{12}$$

where $\widehat{\text{Var}}$, $\widehat{\text{Cov}}$ are respectively the sample variance and sample covariance. If multiple means are estimated per treatment group, then $\hat{V}_{g,g}$ and $\hat{V}_{g,g'}$ will be $k \times k$ matrices instead, but the basic form of (12) remains the same. For parameters that depend on a large number of subsample means, the formula given in (12) may be cumbersome to work with, so a bootstrap approach for standard errors is potentially easier computationally.

2.4 Choice of Machine Learning Estimator

In this section, we give guidance to practitioners about how to choose an ML method for the purposes of performing a flexible regression adjustment. From the perspective of (first-order) asymptotic efficiency, the theory provides relatively simple advice: any ML method that is eventually able to learn the conditional expectation function will yield identical asymptotic properties. Thus, practical advice about the choice of the ML method largely arises from considerations other than asymptotic efficiency.

We begin our discussion by enumerating a number of these practical considerations. First, when we have finite sample considerations it is crucial to recall that some ML methods are highly "data intensive" and hence require large sample sizes to achieve good performance. Second, the ML method should be able to run in a reasonable amount of time; different ML algorithms involve a number of computational steps which depend on the size of the data at different rates and hence, different methods may be more or less costly to run depending on the scale of the experiment being analyzed. Third, the ML method in question should minimize the scope for researcher discretion; ML algorithms with a small number of hyperparameters which can be chosen in an automated manner can help to remove the need for specification search from the researcher and hence mitigate concerns about "p-hacking".

One popular choice of ML method within the economics literature is to use the LASSO

(Urminsky et al. (2016)). One potential explanation for the popularity of the LASSO is that it has good properties with respect to all of the above considerations. By enforcing sparsity, it tends to perform well even in small samples where the sample size, n, is small relative to the dimensionality, p, of X. Its computational complexity also scales well with sample size, and it has a single hyperparameter, the regularization parameter, which can be chosen automatically via cross-validation. One potential pitfall of LASSO, however, is that it requires correct specification of the conditional expectation function as being linear in covariates, X. This concern can be mitigated by attempting to augment X to include basis functions of the underlying covariates, but it may be difficult to do so in a standardized way, which in turn creates a specification search problem.

Motivated by these difficulties with LASSO, in our empirical applications, we instead opt to use tree-based models, such as random forests (Breiman (2001)) or boosting (Freund and Schapire (1997)) instead. Tree-based models confer many of the benefits of LASSO while also covering some of the pitfalls. Specifically, within the ML literature, there exists both formal and informal arguments (Biau (2012), Friedman et al. (2004)) showing that tree-based algorithms implicitly encourage sparse models in a similar fashion as LASSO. This sparsity property implies that tree-based models share many of the small sample benefits of LASSO. Additionally, the recursive partitioning schemes provide a built-in and automated way to detect nonlinearities and interactions present in the data. They therefore do not suffer from specification search problems in the same way that LASSO might. We have also found that boosting in particular is quite computationally scalable as well while random forests perform robustly on small and moderate sized samples.

2.5 Power Calculations

We conclude our theoretical discussion by using our results to provide practical advice for power analyses when using a flexible regression adjustment, as described in this paper. We assume that the researcher has access to pre-exposure analogues of the X's and Y's. In particular, we focus our attention on Equation (3). Recall that for the optimal choice of $\mathbf{h} = \mathbf{m}$, the third term vanishes. Consider now, an estimator of the average treatment effect between two groups g, g'. The variance of the average treatment effect constructed using FRA is given by

$$\frac{1}{n_g}\mathrm{Var}(a_g) + \frac{1}{n_{g'}}\mathrm{Var}(a_{g'}) + \frac{1}{n}\mathrm{Var}(m_g - m_{g'}). \tag{13}$$

Since power calculations are typically conducted to ensure the ability to detect small treatment effects, we make the assumption that treatment effects are "negligible" in the sense that $m_g - m_{g'} \approx 0$ and $\mathrm{Var}(a_g) \approx \mathrm{Var}(a_{g'}) = \mathrm{Var}(a)$. We can therefore take the third term in Equation (13) to be approximately 0. Under these conditions, Equation (13) simplifies to approximately $\left(\frac{1}{n_g} + \frac{1}{n_{g'}}\right) \text{Var}(a)$.

Assume that we have a sample containing pre-exposure analogues of the Y's and X's available to perform power calculations. We now show that we can obtain reasonable estimates of Var(a) using these data. In particular, we observe that by definition

$$\text{Var}(a) = \mathbb{E}\left[\text{Var}(Y|X)\right] = \mathbb{E}\left[(Y - m(X))^2\right], \quad m(x) = \mathbb{E}[Y|X = x].$$

Thus, the asymptotic variance reduction attained by FRA is roughly proportional to the best possible mean-squared error in the population when trying to predict Y given X. Equation (13) shows that the fully regression-adjusted estimator has variance smaller than the variance of the simple difference-in-means estimator by a factor of $\frac{\mathbb{E}[\text{Var}(Y|X)]}{\text{Var}(Y)} \equiv 1 - R^2_{pop}$, or one minus the population R^2 of the nonparametric regression of Y on X. This fact can be helpful when a researcher who plans on using covariates to reduce variance wishes to account for this in the design phase of the experiment.

To see how, consider a typical power calculation, whereby a researcher wishes to calculate the number of observations needed to detect an effect of size M at the $1 - \alpha$ significance level with probability β. If she intends to test this hypothesis with a two-sided t-test from a simple difference in means estimator, the required sample size is roughly

$$N^{SDM} = \frac{2\sigma^2 \left[\Phi(1 - \alpha/2) + \Phi(\beta)\right]^2}{M^2},$$

where σ^2 is the variance of the outcome variable of interest, Y. To perform a power calculation when regression adjustment is anticipated, the correction to the above formula is simply given by

$$N^{FRA} = (1 - R^2)N^{SDM} = (1 - R^2)\frac{2\sigma^2 \left[\Phi(1 - \alpha/2) + \Phi(\beta)\right]^2}{M^2}, \tag{14}$$

where R^2 is the explained variance from the best-fitting nonparametric model predicting Y given X. In a simulation study in the next section, we show that using finite-sample R^2 estimates provides a reasonable guide to estimating the degree of variance reduction in practice as well.

3 Empirical Examples

We now take the theory for flexible regression adjustment derived in the previous section to both simulated and naturally-occurring data. We begin with a number of simulation exercises. First, we construct a synthetic dataset by adding a simulated treatment effect on top of the naturally-occurring data. We then use these simulations to show that the asymptotic theory in Proposition 2 is reliable, but only if proper sample-splitting procedures are followed. This exercise shows that our proposed estimator has good asymptotic properties, but it does not provide much guidance for when one might wish to use an optimal linear regression adjustment (LRA) over a more sophisticated ML-based approach. We thus turn to a number of theory-driven simulation exercises that showcase the salient features of the data that cause ML methods to substantially outperform linear methods.

After showcasing our methods using synthetic datasets, we show that our estimators perform well in naturally-occurring datasets as well. We first turn our attention back to Lyft data and analyze a natural field experiment (see Harrison and List (2004) we conducted at Lyft to show how using regression adjustment reduces variance in a non-negligible manner. In the Lyft setting, we find that although regression adjustments in general make a large difference, the additional flexibility from using an ML model does not substantially improve precision. To explore if ML-based models can yield statistical improvements in other settings, we analyze three additional field experimental datasets where we do find improvements from using a more flexible form of regression adjustment.[10] The four real world applications we consider have substantially varying sample sizes and demonstrate that the flexibility from ML-based approaches can be helpful for various populations of people and situations. Readers interested mainly in the practical implications of our results may skip to Section 3.8, where we summarize our main empirical findings.

Before proceeding, we briefly discuss the choice of machine learning algorithm in our various applications in more detail. Within our Lyft applications, we always use a version of gradient boosting, as implemented in the Python package "XGBoost" (Chen et al. (2015)). In these applications, we typically found that many different hyperparameter choices all gave comparable performance, but good hyperparameters appeared to follow a few basic principles. First, we found that the algorithm tended to perform best when the individual trees within the boosting algorithm are small, with 2 or 3 splits performing best. Second, we found that the algorithm performed best with a low learning rate parameter (in our applications, we always set the learning rate to 0.05) and a large number of boosting rounds. Third,

[10] R Code implementing our flexible regression adjustment along with the analyses of the three non-Lyft settings can be found at the following link: https://github.com/gsun593/FlexibleRA. We have also included a copy of the code in Appendix B

we found that a version of “early stopping” through cross-validation was a computationally simple, yet powerful, way to automatically navigate the bias-variance tradeoff. Specifically, when training the ML model, we always further reserved a small fraction (in our applications 1/8) of our data to serve as a cross-validation dataset. We would then stop updating our XGBoost fit after the model fit on the validation dataset failed to improve for a number of consecutive rounds (in our application, we chose to stop after failing to see improvement for 10 rounds).[11]

Within our non-Lyft data applications, we alternated between using random forests, as implemented in the R package “randomForest” (Liaw et al. (2002)). In general, we found that the random forest algorithm exhibited greater robustness compared to OLS, even with relatively small sample sizes (and, simply using the default hyperparameter settings). This proved important in many of our non-Lyft data examples where the sample sizes are considerably smaller than the Lyft data examples. That said, we found that the boosting algorithm tended to scale more efficiently to large datasets whereas random forests became slow for moderate or large datasets. For sufficiently large datasets (in our examples, we found that $N \geq 10,000$ was a reasonable rule-of-thumb for “sufficiently large”), we found that both algorithms exhibited comparable performance. Thus, in situations with sufficiently large datasets, boosting might be preferred for computational reasons.

3.1 Simulation Study: Augmented Lyft Data

As a baseline, we first check that the asymptotic distribution implied by Proposition 2 approximates the actual sampling distribution of FRA estimators well. To do this, we take a dataset containing one row per registered Lyft passenger.[12] Throughout the exercises in this subsection, we only report the distribution of estimated z-scores using the theory derived in the previous section. Because the z-scores are normalized, they reveal no information about Lyft’s underlying data. However, by comparing their distributions to a standard normal distribution, we can verify that the asymptotic theory we derived in the previous section works well in finite samples.

For each passenger in the dataset, we record as our outcome variable the count of rides that that passenger consumed in a fixed two month window. As covariates, we use a number of summary statistics of past behavior computed on the day before our two month window begins. As one would expect, past behavior tends to be predictive of future behavior, so using

[11] See Friedman et al. (2004) for an interpretation of this strategy as approximating the solution to a LASSO-like estimation procedure.

[12] For confidentiality reasons, we cannot report the exact size of this sample. However, at the time of our writing, Lyft recorded a number of passengers in the tens of millions.

these summary statistics as covariates for regression adjustment is a reasonable approach to reduce variance. We split this dataset randomly into 10,000 smaller datasets and construct "placebo" experiments on these smaller datasets by randomly assigning each individual into "treatment" and "control" with 50/50 probability. For each experiment, we use the sample splitting procedure (with five folds) described in Section 2.3 to estimate the mean potential outcomes in treatment and control. We consider both a "null" case where there is uniformly no treatment effect and an alternative case where we synthetically induce a treatment effect.

To simulate treatment, for each observation i assigned to "treatment," we add a random number of rides drawn from Poisson$(0.1 \cdot r_i)$, where r_i is the number of rides actually attributed to observation i. By construction, the average treatment effect (ATE) from this data generating process is a 10% increase in the number of rides. We construct a point estimate of the ATE by taking the difference of the FRA-adjusted group means. We then use $\hat{V}$ as defined in Equation (12) to compute a standard error estimate. The asymptotics stated in Proposition 2 imply that subtracting the actual treatment effect (0 in the "null" case and $0.1 * mean(\text{rides})$ in the "alternative" case) from the point estimate of the treatment and dividing this difference by the estimated standard error gives us a random variable distributed approximately according to $\mathcal{N}(0, 1)$, so we call these values "z-scores".

In Panels A and B of Figure 2, we plot a histogram of the 10,000 z-scores as well as a qq-plot comparing the empirical quantiles of the z-scores to the normal theoretical quantiles for the dataset with a treatment effect.[13] In Panels A and B of Figure 3, we do the same for the null dataset. In Table 1, we report the mean and variance of the z-scores along with the coverage of 95% and 99% confidence intervals. We find that the z-scores fit the standard normal distribution remarkably well and the coverage of the resulting confidence intervals are statistically indistinguishable from their theoretical values.

We next show the importance of sample splitting. In Panels C and D of Figures 2 and 3, we replicate Panels A and B respectively, except we do not use sample splitting when constructing our estimators. Summary statistics are again in Table 1. Interestingly, when the treatment has a null effect, we find that even without sample splitting, our standard errors provide a reliable approximation to the true sampling distribution of the estimator. However, the situation changes dramatically in the presence of a treatment effect.

For example, examining Figure 2, we see that without sample splitting, the estimates for the average treatment effect with regression adjustment are biased downwards considerably. Intuition about the role of regression adjustment provides a key reason why this bias arises.

[13] Specifically, the x axis in these qq-plots is defined by the theoretical quantiles of a standard normal distribution while the y axis corresponds to the empirical quantiles. If the asymptotic theory is correct, the points in these plots should lie close to the 45 degree line, and deviations from this prediction allow us to more precisely visualize deviations from asymptotic normality.

Recall that FRA estimates treatment group means by adding $\bar{h}_{g,all} - \bar{h}_{g,g}$ to the raw group mean. When $\bar{h}_g$ is estimated in sample, the value of Y_i influences both $\bar{h}_{g,all}$ and $\bar{h}_{g,g}$, but its weight in $\bar{h}_{g,g}$ will be larger. The influence of Y_i on $\bar{h}_g$ tends to push in the opposite direction as its influence in computing $\bar{Y}_g$. This will tend to bias the estimated group means $\hat{\mu}_g$ towards homogeneity, which in turn biases the average treatment effect estimates towards zero. The negative bias we observe in our simulations thus reflects the fact that we constructed a positive treatment effect, so that a bias towards 0 is a negative bias.

Thus far, we have found that not using sample splitting leads to biased estimators, where the bias tends to make group means more similar. One might wonder to what extent a larger sample mitigates this issue. To investigate this question, we repeat the exercises above, but split the data evenly into 1,000 evenly sized datasets, thus increasing the sample size by a factor of 10. We replicate the results of Figures 2 3 in Figures 4 and 5. Summary statistics can again be found in Table 1.

These results make it evident that a large bias continues to exist, even on these larger datasets. Nonetheless, increasing the sample size does appear to attenuate the bias somewhat: while in the moderate sized sample the bias is -1.33 standard errors, in the large sized sample the bias decreases to -1.16.[14] This result suggests that asymptotic unbiasedness might hold, even for the non sample-split estimator, but the asymptote may not be a good approximation, even for large sample sizes. Since cross-fitting does not negatively affect asymptotic efficiency and is usually computationally simple,[15] the evidence presented here supports the conclusion that a practitioner using a flexible regression adjustment should leverage sample splitting for reliable inference.

Proceeding in the opposite direction in terms of sample sizes, one might wonder if flexible regression adjustment can be fruitfully applied on relatively small datasets, which may be the relevant case for many experimental researchers. To address this question, we split our dataset into roughly 100,000 evenly sized smaller datasets, which therefore only have hundreds of observations. We again construct a set of "z-scores" and compare them to the standard normal distribution. We plot the results of this exercise in Figure 6 for the non-null case and 7 for the null case and report summary statistics in Table 1. In this small data regime, a number of additional interesting features are present. First, even the estimator with sample splitting displays a slight amount of bias, although the size of this bias is only about 2% of a standard error. As a result, 95% and 99% confidence intervals

[14] This reduction is not just due to noise: the difference would be statistically significant if subjected to formal hypothesis testing.

[15] If the non-parametric method being used has algorithmic complexity growing faster than linearly in dataset size (which is common), twofold cross-fitting would be even faster than not using a split sample for sufficiently large datasets.

still have excellent coverage properties. Second, examining the null case, both estimators, but especially the non-sample split estimator, display slight deviations from normality, even while the first and second moments appear to closely match their theoretical values. In particular, both estimators display slightly thinned tails (as evidenced by the *over*-coverage of the 95% and 99% confidence intervals), and slightly less mass around values very close to 0 relative to a Gaussian distribution. The thinned tails are likely due to a similar mechanism to the one driving bias in the non-null case: the overfitting from not sample splitting creates a tendency towards mean-reversion, which may be especially effective at correcting extreme cases of imbalance arising due to sampling variability.

Finally, we consider an example where we estimate a quantity that is not an average treatment effect. Consider, for instance, metrics of the form

$$\mathbb{E}[Y_1|W_{i,g}=1]/\mathbb{E}[Y_2|W_{i,g}=1].$$

Two examples of this metric type in a rideshare context are “conversion” (i.e. the probability of taking a ride conditional on opening the app and receiving a price quote and time estimate) and intensive margin labor supply outcomes (i.e. the number of hours a Lyft driver works conditional on working within a given time period). Firms are often interested in learning how this ratio varies in response to different interventions: $\mathbb{E}[Y_1|W_{i,g}=1]/\mathbb{E}[Y_2|W_{i,g}=1]-\mathbb{E}[Y_1|W_{i,g'}=1]/\mathbb{E}[Y_2|W_{i,g'}=1]$. We therefore replicate the exercise in Figure 2 in the context of a ratio metric. Again, we only report z-scores here, which allows us to test the validity of the asymptotic theory while normalizing so as to obscure any identifiable information about Lyft's underlying (confidential) data.

In what follows, for each individual, in addition to examining the number of rides an individual consumes, we also include data on the number of times passengers checked the app, which we call “sessions”. The probability of taking a ride conditional on opening the app is therefore given by the average number of rides divided by the average number of sessions, which takes the form of a ratio metric. We add a treatment effect to this dataset by first adding sessions to each treated individual according to Poisson(s_i), where s_i is 0.05 times the number of sessions taken by individual i. For each added session, we add a ride for that session according to Bernoulli(p_i), where p_i is the proportion of sessions for individual i that resulted in a ride. Finally, for all sessions still not associated with a ride, we add additional rides according to Bernoulli(0.02). The change in conversion in the simulation is therefore $0.02(1-\bar{p})$, where $\bar{p}$ is the population level of conversion in control. We construct point estimates for this quantity by plugging in the regression adjusted means in place of the population means in the formula defining conversion and compute standard errors using

Equation (12) and the delta method.

Empirical results are found in Figure 8, and summary statistics can be found in Table 1. For this particular simulation, we are unable to detect bias in our estimator, but the standard errors are misleadingly small when we do not use sample splitting. As before, the results suggest that sample splitting allows us to do valid inference.

Before closing this subsection, we report the average estimated standard errors from the various simulations performed in Table 2 as a proportion of the standard errors from taking the raw difference in means. There are a number of interesting patterns in this table. In the non sample-split estimators, we find that the within-sample standard errors are considerably smaller than the sample split estimators and become *larger* as a fraction of the difference in means standard error as sample size increases. This fact on its own is unsurprising: for small samples, there is more overfitting while in larger samples, there is less. What is surprising is that the sampling distribution of the "z-scores" suggest that these smaller standard errors end up still being reasonable estimates of the true sampling variability of the estimator. As seen in the non-null data, however, when a treatment effect is present, this reduced sampling variability comes at the cost of large amounts of bias against finding a treatment effect.

A second notable fact is that flexible regression adjustment absorbs a larger proportion of the variability in the data with larger samples. This is a reflection of the fact that non-parametric estimators typically need at least a moderate amount of data to deliver valid results. Interestingly, while we find that in small samples, the non-parametric estimator can deviate considerably from the asymptotic efficiency bound, the plotted distributions of normalized estimates suggest that inference based on such an estimator remains valid. In small data settings, we may therefore prefer to use linear regression adjustment, provided we are confident that a linear functional form is a decent approximation to the conditional expectation function. As discussed in Section 2.4, when data are available ex-ante for doing power calculations, comparing the R^2 from a linear regression to the R^2 from a non-parametric model is a practical way to decide if there is enough data to use a non-parametric method. Moreover, this can be completed prior to examining the experimental results themselves, thus mitigating concerns about specification search or p-hacking.

3.2 Simulation Example: When Does ML Outperform OLS?

While the theoretical results in this paper show that FRA asymptotically can do no worse (and often does better) than the optimal linear regression adjustment (henceforth, LRA), in practice, one may still not always wish to use ML techniques when performing regression adjustment. For example, one reason to prefer LRA is that ML methods can be unstable in small samples and thus deliver less variance reduction than OLS in practice. Another reason to prefer LRA is that ML methods are computationally expensive to use. Fortunately, the efficiency results in Section 2 provides a guide for when FRA is likely to deliver substantial statistical gains relative to LRA.

Specifically, the FRA procedure mainly uses ML techniques to estimate the conditional expectation function of the outcome given the covariates. The LRA, on the other hand, takes the same form as the FRA *except* that the CEF is approximated as linear in covariates. Thus, FRA outperforms LRA when the linear approximation is of low quality. We expect a linear model to poorly approximate the CEF either when the CEF is highly non-linear with respect to each covariate individually or when the CEF contains important interaction effects.

In this section, we conduct a simple family of simulation studies to demonstrate the importance of these two features. Specifically, we simulate treatments $W \sim \text{Bernoulli}(0.5)$ as well as three latent variables L_1, L_2, L_3 with $L_1, L_2, L_3 \sim \text{Unif}(0,1)$ which are unobserved to the econometrician. We also simulate an unobserved "error term" $U \sim \mathcal{N}(0,1)$. Our outcome Y is then given by

$$Y = W + L_1 + L_2 + L_3 + U$$

In three of our simulations, we derive our covariates X_1, X_2, X_3 from L_1, L_2, L_3 according to $X_i = L_i^p$ for $p = 1, 5, 10$. In addition, we consider three additional specifications where $X_1 = (L_1 L_2)^p$, $X_2 = (L_2 L_3)^p$ and $X_3 = (L_3 L_1)^p$ for $p = 1, 5, 10$ to generate non-trivial interaction terms. For each simulation, we draw $B = 100$ samples of size $N = 1,000$.

We specified our simulations so that the true signal/noise ratio is identical across specifications, but as p increases further from 1, the true CEF becomes increasingly non-linear. In Table 3, we report the ratio of the standard errors from FRA compared to LRA in the 6 specifications (3 values of p times 2 specifications varying whether or not there are interactions amongst covariates). As expected, as p increases away from 1 or as we add an interaction amongst covariates, the gains from FRA over LRA increase. Note however, that when there is no interaction and $p = 1$, FRA performs slightly worse than LRA. Asymptotically, because the CEF is linear, we expect FRA and LRA to attain the same sized standard errors. However, in finite sample sizes, we expect FRA in this case to perform worse due to the fact that machine learning techniques tend to require more data to achieve a good fit. This shows that when the true CEF is known to be linear (or approximately linear), LRA may still be preferable over FRA in practice, even ignoring computational considerations.

3.3 Simulation Example: Variance Reduction and R^2

We now use the same simulation setup as above to probe the reliability of Equation (14) in estimating the amount of variance reduction one should expect when using regression adjustment. One issue that was left somewhat unspecified in (14) is the relevant R^2 to plug into the power calculation formula. In this section, we show that *finite-sample, cross-validation* R^2 provides a reliable, albeit somewhat conservative guide in practice. For each value of $p \in \{1, 5, 10\}$ and $N \in \{30, 60, 90, 120, 150\}$, we simulate 100 datasets according to the same data generating process as in the data-generating process outlined in the previous section. On each simulated dataset, we compute the following:

1. Compute simple difference in means and FRA estimates of treatment effects with standard errors. Call the standard errors respectively $\hat{\sigma}_s$ and $\hat{\sigma}_f$.

2. Compute the mean-squared error of the ML estimate used to construct the FRA estimates in predicting outcomes in the relevant treatment group. Call this $\widehat{MSE}$.

3. Compute the variance of the outcome. Call this $\hat{\sigma}_Y$.

Recall that the ML predictions for any given observation i derived in step 2 above only uses data that does not include observation i. Thus, step 2 measures the finite-sample, cross-validation mean-squared error of the fitted ML models used for regression adjustment. Our conjecture, motivated by a finite-sample analogue of Equation (14), is therefore that

$$\frac{\hat{\sigma}_f^2}{\hat{\sigma}_s^2} \approx \frac{\widehat{MSE}}{\hat{\sigma}_Y^2} \equiv 1 - \hat{R}^2.$$

To test this conjecture, for each fixed value of p and N, we compute the average value of $\frac{\hat{\sigma}_f^2}{\hat{\sigma}_s^2}$ and $1 - \hat{R}^2$ across the 100 simulation draws and plot the results in Figure 9.

Fixing a single value of p (i.e. fixing the DGP), we see that the finite-sample performance of the ML model improves as sample sizes increase, in the sense that $1 - \hat{R}^2$ gets smaller the smaller is N. We see that the variance reduction attained by FRA estimator shrinks roughly one-for-one with the improvements in $1 - \hat{R}^2$, although the actual variance reduction across the board is even larger than $1 - \hat{R}^2$. This suggests that $1 - \hat{R}^2$ is if anything a conservative estimate of actual variance reduction. Overall, these results suggest that $\hat{R}^2$ as defined here is a reasonable quantity to plug into Equation (14) for the purposes of power calculation.

3.4 Application I: Natural Field Experiment on Lyft Cancellations

Having shown that cross-fitting allows us to robustly quantify uncertainty in a simulation setting, we apply the FRA to a natural field experiment. The premise behind our field experiment is that, ceteris paribus, Lyft prefers to minimize the number of cases wherein a driver agrees to pick up a passenger, but then cancels the ride before pickup occurs. Formally, Lyft would like to design a policy to minimize the cancellation rate, $\frac{\frac{1}{n}\sum_{i=1}^{n} \# \text{ rides canceled}}{\frac{1}{n}\sum_{i=1}^{n} \# \text{ rides accepted}}$. When considering a new intervention, Lyft would therefore like to track how this metric varies across different treatment conditions.

Here, we focus on a field experiment we helped to conduct at Lyft in 2018. To ensure that drivers do not waste time waiting for passengers who never show up, Lyft allows drivers the option to mark a passenger as a "no show" if sufficient time elapses after the driver arrives at the pickup location (this is considered a special type of cancellation). Passengers who are marked as "no-show" are charged a fee which is passed on to the driver to compensate for lost passenger time. When Lyft introduced its Shared rides (a product whereby a passenger receives a discount in return for allowing Lyft to match them with another passenger taking a similar route at the same time), it had to rethink its original no-show policy. Specifically, because of the fact that multiple passengers potentially shared the same driver, a passenger not promptly arriving to their pickup location would impose a negative externality on passengers already in the car. As a result, Lyft decided that the window of time passengers received before the driver was allowed to mark them as a "no show" for Shared rides should be less generous. This led to the rate of no-show cancels to be higher on Shared rides relative to Standard rides.

Before our field experiment, the status quo policy was that *all* Shared rides had a shorter no-show window. However, such a uniform policy was irrational if a passenger requesting a Shared ride was matched to a driver without other passengers already in the car, since the negative externality is not present. This reasoning represents the genesis of our experiment. In our field experiment, drivers were assigned to treatment or control. Control drivers received the status quo policy whereas if a passenger was matched to a treated driver and was the first passenger in the car, they would receive the more generous no-show window that Standard passengers received.

Our two key outcome metrics are the cancellation rate, as defined above, and the no show rate, defined as $\frac{\frac{1}{n}\sum_{i=1}^{n} \# \text{ rides canceled because no show}}{\frac{1}{n}\sum_{i=1}^{n} \# \text{ rides accepted}}$. We fit three models. First, we consider simply plugging the subsample means (SM) within each variant into the formulae defining the cancellation and no show rates. Second, we consider plugging in the linear regression adjusted estimates.[16] Finally, we apply the fully flexible regression adjustment (FRA). Standard errors are constructed using the delta method. For each metric, we report the point estimate of the effect as a percent of the baseline in control and the standard errors.

A summary of our empirical results are reported in Table 4. A number of notable facts stand out. First, the experiment was a success when considering the no-show rate, which decreased by roughly 4.5% across estimators; the null hypothesis of no effect is easily rejected. Second, despite this, when examining the treatment effect on the overall cancellation rate, we have difficulty detecting a significant effect with the non-regression adjusted estimate. Yet, reducing variance using a regression adjustment (either LRA or FRA) makes the effect considerably easier to detect. Third, while it is true that, as the theory predicts, a fully flexible regression adjustment delivers slight efficiency gains over a linear adjustment, these gains appear to be modest in practice.

In Table 5, we investigate this further by reporting the R^2 (defined as one minus the out-of-sample mean squared prediction error divided by total variance) of the linear model compared to the non-parametric model in explaining variation in the number of accepts, number of cancels, and number of no shows. Indeed, for most of the outcomes of interest studied here, the non-parametric models appear to explain only slightly more of the variation than a linear model, and even provides a slightly worse fit for predicting the number of cancels. The limited additional gains from using the flexible regression adjustment in our setting likely implies that the conditional expectation functions for the outcomes we are examining are reasonably well approximated by linear functions.

We conclude this section by also comparing regression adjustments to the TWFE model. For each individual, in addition to the number of acceptances, cancels, and no shows during the experimental period, we also obtain data about the number of acceptances, cancels, and no shows in the period before the experiment started. In Table 6, we report the degree to which differences-in-differences reduces variance compared to a regression adjustment estimator that simply takes these pre-experimental outcomes as covariates.

Depending on the outcome we are examining, we find that regression adjustment results in a 1-10% reduction in the size of the standard errors. Moreover, the square of the ratios reported in Table 6 represent how much smaller the sample size needs to be holding statistical power fixed if one uses regression adjustment instead of differences-in-differences. Across our three outcomes, we find that regression adjustment allows an experimenter to garner the same power for a sample with only 84% to 97% of the number of observations, suggesting that the current practice among experimental economists of estimating TWFE models may be causing

[16] Specifically, we implemented our point estimates according to 11 and our standard errors according to 12, but using an OLS fit for $\hat{m}_{g,i}$ in place of a fitted machine learning model.

researchers to “overpay” for their experiments by a non-trivial amount, leading to a greater number of Type 2 errors. This result suggests that experimental economists commitment to focusing solely on sample size or sample allocation across cells when considering power is unduly restrictive, and indeed quite inefficient.

3.5 Application II: Oregon Health Insurance Experiment

We next turn our attention to an analysis of the data from the Oregon Health Insurance Experiment (OHIE). We focus in particular on replicating the results of Finkelstein et al. (2016), which measures the impact of Medicaid on emergency room visits. We take our covariates to be gender, age, prior health, and education along with a detailed vector of counts for various types of ER visits prior to randomization. Our outcome of interest is whether or not an individual visited an emergency department during the experiment. We additionally estimate the impact of treatment status on medicaid take-up, and by dividing the reduced form effect of treatment on outcome by the effect of treatment on take-up, we can estimate the LATE of Medicaid take-up on ER visits.

Empirical results of this exercise are summarized in Table 7, which has a similar form as Table 4 and compares the subsample means estimator which does not use any covariates to the linear and flexible adjustments. Across specifications, we find that the flexible regression improves standard errors by about 2-3% relative to the next best alternative. While these gains are modest fixing sample size, they imply that for a similar level of statistical power, researchers could reduce sample sizes by about 5-6%, thus reducing variable experimental costs by a similar quantity. We suspect that with a richer covariate set the gains would have been even greater.

3.6 Application III: Water Conservation Nudges

We next re-analyze data from a natural field experiment conducted by Ferraro and Price (2013), which studies the effect of a number of nudges on water conservation. The intervention was designed to reduce water consumption during the summer months of 2007. In addition to collecting data on summer consumption, the authors also collect month-by-month water consumption for each individual in the experiment in the year prior to experimentation.

For simplicity, we only consider the effect of their strongest nudge treatment relative to the control group. We consider three sets of analyses, with results displayed in Table 8. First, we replicate the basic specification of Ferraro and Price (2013), with the small difference that we further disaggregate their pre-intervention measures of water use and include a separate covariate for each month. Our outcome, Y, measures levels of water consumption in June, July, August, and September of 2007. Our covariates, X, consisted of a vector of monthly levels of water consumption for each household in the year prior to the intervention.

With this specification, ML delivers similar levels of precision improvements as in the OHIE, reducing standard errors by 3% and implying that the sample size could be reduced by 6% while holding statistical power fixed. When replicating their results, we noticed that the distribution of outcomes is fairly skewed, so we next considered a specification where we instead took our outcome to be $\log(Y + 1)$. In this specification, we found substantial gains to using an ML technique. Relative to the linear specification, ML reduced standard errors by 13%, which equivalently can be thought of as implying that a sample size reduction of 24% would leave statistical power unchanged.

An important reason for this discrepancy is that while outcomes are measured in logs, the covariates in the second specification continued to be expressed as levels. We thus consider a third specification where we measure covariates X in logs as well, $\log(X + 1)$. Under this specification, the gains to using an ML technique once again look modest relative to the linear specification. Standard errors are smaller by roughly 2%, or equivalently, sample size could be reduced by roughly 4% holding power fixed. We view this example as demonstrating an important methodological point. If the researcher has good intuition about the functional relation between outcomes and covariates, there are limited gains to using ML techniques over a well-specified linear regression. In this particular case, it is fairly intuitive that if outcomes are logged, then corresponding covariates should be measured in logs as well. However, our example shows that that ML-based regression adjustments are considerably more robust to pre-analysis transformations that the researcher might make to covariates and thus may be especially helpful when the researcher does not have strong prior information about which functional form specifications are most likely to be accurate.

3.7 Application IV: CogX, An Early Education Program

Our final empirical example is the evaluation of data from the CogX program described in Fryer et al. (2020). The experiment studies the effect of an early childhood intervention, CogX, on cognitive and non-cognitive test scores. Following the authors, we focus in particular on the effect of CogX on an index of cognitive test scores. For controls, we include a number of demographic variables (birth weight, mother's education, mother's age, household income, race, and gender) as well as pre-intervention test scores.

We include a summary of empirical results in Table 9. We find that ML techniques are able to reduce standard errors by roughly 4% in this setting, which alternatively implies that sample size could have been reduced by roughly 8% while maintaining statistical power. In this setting, this would have amounted to hundreds of thousands of dollars.

3.8 Summary of Empirical Results

In this section, we synthesize the key insights from our empirical exercises:

1. FRA is reliable, but only if one uses cross-fitting
    - Even at relatively small sample sizes, FRA estimates are approximately unbiased.
    - Standard errors conform to the asymptotic formula of Equation (12).
2. FRA reduces standard errors by more than LRA across a wide variety of settings. In panel settings, FRA also outperforms fixed-effects based estimators.
3. The advantages of FRA are larger when interactions and nonlinearities are likely to be important features of the CEF.
4. Finite-sample, cross-validated R^2 in predicting Y using X corresponds roughly to the degree of variance reduction attainable in practice.
5. FRA helps minimize the need for ad hoc specification search
    - In some settings, the performance of LRA may be highly sensitive to how the variables in X are transformed prior to regression.
    - The performance of FRA is less sensitive to how X is transformed.
6. Tree-based algorithms perform well in general in our data sets. Which algorithm to choose may depend on sample size:
    - Random forests perform well across sample sizes, even with default hyperparameter settings, but may be slow for larger sample sizes.
    - Boosting is reliable at sample sizes where random forests become prohibitively expensive, which we find to be around $n \approx 10,000$. In those settings, using trees with low depth (2-3 splits), low learning rate (we typically set our learning rate to 0.05), and early stopping appears to produce robust performance with minimal researcher input.

4 Conclusion

In this paper, we synthesize and generalize a number of approaches to reducing variance when analyzing experimental data. We expand on prior theory along a number of dimensions. We consider a broad class of regression adjustment estimators and identify the conditional expectation function as the minimum variance function to use for the adjustment. We then show that regression adjustment estimators can be written in a way that satisfies an *orthogonality* property, which in turn makes it possible to use non-parametric/machine learning estimators to implement feasibly the optimal adjustment under mild regularity assumptions.

These efficiency and feasibility results have important implications for researchers designing and analyzing experiments. They suggest that provided a good approximation to the conditional expectation function is being used for variance reduction, there are limited gains to using additional clever econometrics to enhance precision in experimental data. Importantly, regardless of the parameter being estimated, our results suggest that a researcher seeking a greater level of experimental power should focus more on finding better covariates than on finding better estimators.

We also showcase the practical implications of our theoretical results in a number of synthetic and naturally-occurring datasets. We begin by performing a number of simulation studies by augmenting naturally-occurring Lyft data on outcomes and covariates with synthetic treatments. We show that across a range of sample sizes from hundreds to hundreds of thousands, our asymptotic results provide a reliable guide to inference. We then construct a set of additional simulations to show when researchers may wish to use ML vs linear regression adjustment techniques. After running these simulation studies, we turn towards the analysis of a number of real-world datasets where we are able to quantify the performance various regression adjustment techniques.

Our empirical examples provide a number of key takeaways. First, flexible regression adjustment is not solely a technique for the big data world. Even with relatively small datasets with only hundreds of observations, FRA can improve precision meaningfully. Second, our simulations suggest that sample splitting is crucial for ensuring that the standard errors from the regression adjustment are valid, but once sample splitting is used, inference is reliable and tractable. Third, our example using Lyft data gives a real-world example where a currently popular applied practice of analyzing field experiments with pre-treatment outcomes using two-way fixed effects estimators can lead to substantial losses in statistical precision relative to regression adjustment. This empirically supports our theoretical result that the two-way fixed effects estimators are statistically inefficient. They should therefore typically be replaced with some form of regression adjustment, linear or otherwise. Fourth, in a number of non-Lyft datasets, we find that the additional gains to using ML techniques over linear regression adjustment can allow researchers to attain similar levels of statistical power with 4-8% fewer observations.

While the results we present are able to generalize the existing literature in a number of ways, they are suggestive of some avenues for future work, which we briefly discuss. First, our efficiency results only apply to the case of i.i.d. sampling. Future work should explore whether the efficiency of our proposed estimator is robust to alternative experiment designs such as blocking or stratification. Second, our results only apply to the cases where the parameters of interest can be expressed as functions of a finite set of sample means. As aforementioned, in some settings, researchers are interested in using experimental variation to estimate a structural parameter. When the estimator of the structural parameter of interest is obtained by optimizing a criterion function, our results may not directly apply. We are currently exploring the possibility of generalizing the ideas behind this present work to that setting.

References

Armstrong, T. B. (2022). Asymptotic efficiency bounds for a class of experimental designs. *Working Paper*.

Biau, G. (2012). Analysis of a random forests model. *The Journal of Machine Learning Research*, 13:1063–1095.

Bodoh-Creed, A. L., Hickman, B. R., List, J. A., Muir, I., and Sun, G. K. (2023). Stress testing structural models of unobserved heterogeneity: Robust inference on optimal nonlinear pricing. Technical report, National Bureau of Economic Research.

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., et al. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68.

Cohen, P. L. and Fogarty, C. B. (2023). No-harm calibration for generalized oaxaca-blinder estimators. *Biometrika.*

Cotton, C., Hickman, B., List, J., Price, J., and Roy, S. (2020). Productivity versus motivation: Combining field experiments with structural econometrics to study adolescent human capital production. *Working paper.*

DellaVigna, S., List, J. A., and Malmendier, U. (2012). Testing for altruism and social pressure in charitable giving. *The Quarterly Journal of Economics*, 127:1–56.

Duflo, E., Hanna, R., and Ryan, S. P. (2012). Incentives work: Getting teachers to come to school. *American economic review*, 102(4):1241–1278.

Ferraro, P. J. and Price, M. K. (2013). Using nonpecuniary strategies to influence behavior: evidence from a large-scale field experiment. *Review of Economics and Statistics*, 95(1):64–73.

Finkelstein, A. N., Taubman, S. L., Allen, H. L., Wright, B. J., and Baicker, K. (2016). Effect of medicaid coverage on ed use—further evidence from oregon's experiment. *New England Journal of Medicine (NEJM/MMS).*

Fisher, R. (1935). *The Design of Experiments.* Oliver and Boyd.

Fowlie, M., Wolfram, C., Spurlock, C. A., Todd, A., Baylis, P., and Cappers, P. (2020). Default effects and follow-on behavior: Evidence from an electricity pricing program. *Working Paper.*

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Friedman, J., Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. (2004). Discussion of boosting papers. *Annual Statistics*, 32:102–107.

Frison, L. and Pocock, S. (1992). Repeated measures in clinical trials: Analysis using mean summary statistics and its implications for design. *Statistics in Medicine*, 11:1685–1704.

Fryer, Jr., R. G., Levitt, S. D., List, J. A., and Samek, A. (2020). Introducing cogx: A new preschool education program combining parent and child interventions. Technical report, National Bureau of Economic Research.

Glynn, A. N. and Quinn, K. M. (2010). An introduction to the augmented inverse propensity weighted estimator. *Political analysis*, 18(1):36–56.

Goldszmidt, A., List, J., Metcalfe, R., Muir, I., Smith, V. K., and Wang, J. (2021). The value of time in the united states: Estimates from nationwide natural field experiments. *Working Paper*.

Goodman-Bacon, A. (2021). Difference-in-differences with variation in treatment timing. *Journal of Econometrics*.

Gosnell, G., List, J., and Metcalfe, R. (2020). The impact of management practices on employee productivity: A field experiment with airline captains. *Journal of Political Economy*, 128(4):1195–1233.

Guo, Y., Coey, D., Konutgan, M., Li, W., Schoener, C., and Goldman, M. (2021). Machine learning for variance reduction in online experiments. *Advances in Neural Information Processing Systems*, 34:8637–8648.

Györfi, L., Kohler, M., Krzyżak, A., and Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression*. New York: Springer-Verlag.

Hahn, J. (1998). On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica*, 66(2):315–331.

Harrison, G. W. and List, J. A. (2004). Field experiments. *Journal of Economic Literature*, 42(4):1009–1055.

Jin, Y. and Ba, S. (2023). Toward optimal variance reduction in online controlled experiments. *Technometrics*, 65(2):231–242.

Kaplan, S., Moskowitz, T., and Sensoy, B. (2013). The effects of stock lending on security prices: An experiment. *Journal of Finance*, 68(5):1891–1936.

Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

Negi, A. and Wooldridge, J. (2020). Robust and efficient estimation of potential outcome means under random assignment. *Working Paper*.

Negi, A. and Wooldridge, J. (2021). Revisiting regression adjustment in experiments with heterogeneous treatment effects. *Econometric Reviews*, 40(5):504–534.

Opper, I. M. (2021). Improving average treatment effect estimates in small-scale randomized controlled trials. In *EdWorkingPaper: 21–344*.

Pitkin, E., Berk, R., Brown, L., Buja, A., George, E., Zhang, K., and Zhao, L. (2013). Improved precision in estimating average treatment effects. *arXiv preprint arXiv:1311.0291*.

Poyarkov, A., Drutsa, A., Khalyavin, A., Gusev, G., and Serdyukov, P. (2016). Boosted decision tree regression adjustment for variance reduction in online controlled experiments. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 235–244.

Rosenblum, M. and Van Der Laan, M. J. (2010). Simple, efficient estimators of treatment effects in randomized trials using generalized linear models to leverage baseline variables. *The international journal of biostatistics*, 6(1).

Roth, J. and Sant'Anna, P. H. (2023). Efficient estimation for staggered rollout designs. *Journal of Political Economy: Microeconomics*.

Rothe, C. (2020). Flexible covariate adjustments in randomized experiments. *Working Paper*.

Spiess, J. (2018). Optimal estimation when researcher and social preferences are misaligned.

Todd, P. and Wolpin, K. (2006). Assessing the impact of a school subsidy program in mexico: Using a social experiment to validate a dynamic behavioral model of child schooling and fertility. *American Economic Review*, 96(5):1384–1417.

Tsiatis, A. A., Davidian, M., Zhang, M., and Lu, X. (2008). Covariate adjustment for two-sample treatment comparisons in randomized clinical trials: a principled yet flexible approach. *Statistics in medicine*, 27(23):4658–4677.

Urminsky, O., Hansen, C., and Chernozhukov, V. (2016). Using double-lasso regression for principled variable selection. *Available at SSRN 2733374*.

Wager, S., Du, W., Taylor, J., and Tibshirani, R. (2016). High-dimensional regression adjustments in randomized experiments. *Proceedings of the National Academy of Sciences*, 113(45):12673–12678.

Wu, E. and Gagnon-Bartsch, J. A. (2018). The loop estimator: Adjusting for covariates in randomized experiments. *Evaluation review*, 42(4):458–488.

Zhang, M., Tsiatis, A. A., and Davidian, M. (2008). Improving efficiency of inferences in randomized clinical trials using auxiliary covariates. *Biometrics*, 64(3):707–715.

A Proofs

A.1 Proof of Proposition 1

The potential outcome means $\boldsymbol{\mu}$ can be characterized as solutions to the following moment condition:

$$\sum_{i=1}^{N} Y_i - \boldsymbol{\mu}_i' \mathbf{W}_i = 0$$

Zhang et al. (2008) shows that any unbiased estimator $\tilde{\boldsymbol{\mu}}$ of $\boldsymbol{\mu}$ making use of information in $(Y_i, \mathbf{W}_i, X_i)$ can be represented as a vector such that the g^{th} component of $\tilde{\boldsymbol{\mu}}$ is given by the solution to the following moment condition:

$$\sum_{i=1}^{N} \left\{ (Y_i - \tilde{\mu}_g)\mathbb{1}\{W_{i,g} = 1\} + \sum_{g'=1}^{G} (\mathbb{1}\{W_{i,g'} = 1\} - \rho_{g'}) h_g^{g'}(X_i) \right\}$$

Rearranging, this implies that

$$\tilde{\mu}_g^{h_g^1,\ldots,h_g^G} \equiv \bar{Y}_g + \sum_{g'} \bar{h}_{g,all}^{g'} - \bar{h}_{g,g'}^{g'}$$

for some functions $h_1, \ldots, h_G$. Note that this is a similar form as in (1), but now, the estimator for the mean potential outcome in group g may also contain adjustment functions for groups $g' \neq g$, while (1) contains an adjustment function only for group g.

We show that for any such estimator, $\text{Var}(\hat{\mu}_g^{h_1,\ldots,h_G}) \geq \text{Var}(\hat{\mu}_g^{h_g})$. The derivation of Equation (3) in the main body can be used to show that

$$\tilde{\mu}_g^{h_g^1,\ldots,h_g^G} = \underbrace{\bar{\varepsilon}_g}_{\equiv \mathbf{A}} + \underbrace{\bar{m}_{g,all}}_{\equiv \mathbf{B}} + \underbrace{(\bar{d}_{g,all}^g - \bar{d}_{g,g}^g) + \sum_{g' \neq g} \bar{h}_{g,all}^{g'} - \bar{h}_{g,g'}^{g'}}_{\equiv \mathbf{C}^{\mathbf{h}}}$$

The derivations in the main body of the paper imply that $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}^{\mathbf{h}}$ are uncorrelated with one another. So variance is minimized when the variance of $\mathbf{C}^{\mathbf{h}}$ is minimized. The variance of this third term is set to 0 when $h_g^g = m_g$ and when $h_g^{g'} = 0$ for any $g' \neq g$, which is exactly the efficient estimator derived in the main body of the paper.

A.2 Proof of Lemma 1

Note that constant shifts in h_1, h_0 do not affect the resulting estimator, so WLOG, we may take h_1, h_0 so that $\mathbb{E}[d_g(X_i)] = 0$ for $g = 0, 1$. From the text, note that $\text{Var}(\hat{\mu}_1^{h_1} - \hat{\mu}_0^{h_0}) -$

$\mathrm{Var}(\hat{\mu}_1^{m_1} - \hat{\mu}_0^{m_0}) = \mathrm{Var}((\bar{d}_{1,all} - \bar{d}_{1,1}) - (\bar{d}_{0,all} - \bar{d}_{0,0}))$. We can write expression in the variance on the RHS out explicitly as

$$RHS = \frac{1}{N}\sum_{i=1}^{N}\left[d_1(X_i)\left(1 - \frac{N}{N_1}W_i\right) + d_0(X_i)\left(1 - \frac{N}{N_0}(1 - W_i)\right)\right].$$

Rearranging, we have that

$$RHS = -\frac{1}{N}\sum_{i=1}^{N}\left[W_i\left(\frac{N_0}{N_1}d_1(X_i) + d_0(X_i)\right) + (1 - W_i)\left(\frac{N_1}{N_0}d_0(X_i) + d_1(X_i)\right)\right].$$

Note that as $N \to \infty$, $N_0/N_1 \xrightarrow{p} \rho_0/\rho_1$, so we may replace N_0/N_1 with ρ_0/ρ_1 in the above and obtain an asymptotically equivalent expression. Taking variances of the above expression, using the law of total variance, and using the fact that observations are i.i.d., we have

$$N\mathrm{Var}(RHS) = \rho_1 \mathrm{Var}\left(\frac{\rho_0}{\rho_1}d_1(X_i) + d_0(X_i)\right) + \rho_0 \mathrm{Var}\left(\frac{\rho_1}{\rho_0}d_0(X_i) + d_1(X_i)\right)$$

Expanding out these variances shows that

$$N\mathrm{Var}(RHS) = \rho_0\left(1 + \frac{\rho_0}{\rho_1}\right)\mathrm{Var}(d_1(X_i)) + \rho_1\left(1 + \frac{\rho_1}{\rho_0}\right)\mathrm{Var}(d_0(X_i)) + 2\mathrm{Cov}(d_1(X_i), d_0(X_i))$$

Further algebraic manipulations using the fact that $\rho_1 + \rho_0 = 1$ shows that

$$N\mathrm{Var}(RHS) = \frac{\rho_0^2\mathrm{Var}(d_1(X_i)) + \rho_1^2\mathrm{Var}(d_0(X_i)) + 2\rho_0\rho_1\mathrm{Cov}(d_0(X_i), d_1(X_i))}{\rho_0\rho_1}$$

This final expression simplifies to the desired conclusion that

$$N\mathrm{Var}(RHS) = \frac{\mathrm{Var}(\rho_0 d_1(X_i) + \rho_1 d_0(X_i))}{\rho_0\rho_1}$$

B Code for Non-Lyft Analyses

```
library(dplyr)
library(gbm)
library(randomForest)
library(numDeriv)
library(ggplot2)


# Perform Flexible Regression Adjustment Pre-Processing
# FRA(dat, outcome_cols, treat_col, covariate_cols, n_folds, method)
# Inputs:
#   dat: data frame with outcomes, treatments, and covariates
#   outcome_cols: column names for outcomes of interest
#   treat_col: column name of treatment
#   covariate_cols: column names of covariates
#   n_folds: number of folds for sample splitting
#   method: regression method used for regression adjustment
#   ML_func: Custom ML model supplied by user. Should be of the form ML_func(formula, data).
#            Output should have a predict function.
#
# Output:
#   dat_with_FRA: original dataframe with extra columns of the form
#       'm_{otcuome name}_{treatment name}': fitted value of conditional expectation,
#         E[outcome | X, treatment] for the outcome and treatment named
#       'u_{outcome name}_{treatment name}': "influence function" for mean potential outcome,
#         E[outcome(treatment)]. Mean of this column is the regression adjusted estimator for
#         E[outcome(treatment)] and variance-covariance matrix of these columns is asymptotically
#         valid estimator of covariance matrix of the regression adjusted point estimates
#####
FRA <- function(dat, outcome_cols = c('Y'),
                     treat_col = 'W',
                     covariate_cols = c('X1', 'X2', 'X3'),
                     n_folds = 2,
                     method = '',
                     ML_func = NULL, num_trees = 300) {
  # Split sample to ensure balance in treatment status across samples
  dat <- dat %>% as.data.frame
  dat$order <- sample(1:nrow(dat), nrow(dat))
  dat <- dat %>% arrange(!!sym(treat_col), order)
  fold_col <- rep(1:n_folds, ceiling(nrow(dat) / n_folds))
  fold_col <- fold_col[1:nrow(dat)]
  dat$fold <- fold_col

  # Get unique treatment levels
  treat_levels <- unique(dat[,treat_col]) %>% as.vector


  # Perform Crossfitting
  # Split out by method
  # For each outcome/treatment pair, create column called 'm_{outcome name}_{treatment name}'
  # which is the best predictor of outcome given covariates within treatment group
  if (method == 'linear') {
    for (y in outcome_cols) {
      for (treat in treat_levels) {
        # Create new column for m_{outcome name}_{treatment name}
        dat[,paste('m_', y, '_', treat, sep = '')] <- 0
        for (f in 1:n_folds) {
          # Fit OLS model using data from folds except current fold
          lmod <- lm(formula(paste(y, '~', paste(covariate_cols, collapse = '+'))),
                     dat %>% filter(f != fold, !!sym(treat_col) == treat))
          # Project fitted values based on covariates of current fold
          dat[dat$fold == f,paste('m_', y, '_', treat, sep = '')] <- predict(lmod, dat %>%
                                                                              filter(fold == f))
        }
      }
    }
  }
  else if (method == 'rf') {
    for (y in outcome_cols) {
      for (treat in treat_levels) {
        # Create new column for m_{outcome name}_{treatment name}
        dat[,paste('m_', y, '_', treat, sep = '')] <- 0
        for (f in 1:n_folds) {
          # Fit random forest model using data from folds except current fold
          rfMod <- randomForest(formula(paste(y, '~', paste(covariate_cols, collapse = '+'))),
                     dat %>% filter(f != fold, !!sym(treat_col) == treat))
          # Project fitted values based on covariates of current fold
          dat[dat$fold == f,paste('m_', y, '_', treat, sep = '')] <- predict(rfMod, dat %>%
                                                                              filter(fold == f))
        }
      }
    }
  }
  else if (method == 'gbm') {
    for (y in outcome_cols) {
      for (treat in treat_levels) {
        # Create new column for m_{outcome name}_{treatment name}
        dat[,paste('m_', y, '_', treat, sep = '')] <- 0
        for (f in 1:n_folds) {
          # Fit gradient boosting machine model using data from folds except current fold
          gbmMod <- gbm(formula(paste(y, '~', paste(covariate_cols, collapse = '+'))),
                               dat %>% filter(f != fold, !!sym(treat_col) == treat),
                      interaction.depth = 2, n.trees = num_trees, shrinkage = 0.05,
                      distribution = 'gaussian', verbose = F)
          # Project fitted values based on covariates of current fold
          dat[dat$fold == f,paste('m_', y, '_', treat, sep = '')] <- predict(gbmMod, dat %>%
                                                                             filter(fold == f))
        }
      }
    }
  }
  else if (!is.null(ML_func)) {
    for (y in outcome_cols) {
      for (treat in treat_levels) {
        # Create new column for m_{outcome name}_{treatment name}
        dat[,paste('m_', y, '_', treat, sep = '')] <- 0
        for (f in 1:n_folds) {
          # Fit OLS model using data from folds except current fold
          ML_mod <- ML_func(formula(paste(y, '~', paste(covariate_cols, collapse = '+'))),
                    dat %>% filter(f != fold, !!sym(treat_col) == treat))
          # Project fitted values based on covariates of current fold
          dat[dat$fold == f,paste('m_', y, '_', treat, sep = '')] <- predict(ML_mod, dat %>%
                                                                             filter(fold == f))
        }
      }
    }
  }
  else {
    stop("Method most be in c('linear', 'rf', 'gbm') or custom method must be supplied")
  }

  # For each outcome/treatment pair, create column for influence function of the form
  # 1 / prob(treatment) * (Y - E[Y|X,treatment]) * 1{treatment} + E[Y|X,treatment]
  for (treat in treat_levels) {
    prop_treat <- mean(dat[,treat_col] == treat)
    for (y in outcome_cols) {
      dat <- dat %>% mutate(
        !!sym(paste('u_', y, '_', treat, sep = '')) :=
          case_when(!!sym(treat_col) == treat ~ 1/prop_treat *
                     (!!sym(y) - !!sym(paste('m_', y, '_', treat, sep = ''))),
                   TRUE ~ 0) + !!sym(paste('m_', y, '_', treat, sep = ''))
      )
    }
  }
  dat_with_FRA <- dat
  dat_with_FRA
}
#####


# Estimate Average Treatment Effect after Full Regression Adjustment Pre-processing
# FRA_ATE(dat_with_FRA, outcome_col, treat_lvl, ctrl_lvl)
# Inputs:
#   dat_with_FRA: dataframe with regression adjusted columns
#   outcome_col: name of outcome whose ATE is being estimated
#   treat_lvl: value of W corresponding to "treatment"
#   ctrl_lvl: value of W corresponding to "control"
#
# Output:
#   Vector with point estimate and standard error
#####
FRA_ATE <- function(dat_with_FRA, outcome_col = 'Y', treat_lvl, ctrl_lvl) {
  tmp <- dat_with_FRA %>%
    mutate(u = !!sym(paste('u_', outcome_col, '_', treat_lvl, sep = '')) -
             !!sym(paste('u_', outcome_col, '_', ctrl_lvl, sep = '')))

  c(tmp %>% .$u %>% mean, (tmp %>% .$u %>% sd) / sqrt(nrow(tmp)))
}
#####
```

```
# Estimate local average treatment effect when experiment assignment W is instrument for treatment
# FRA_LATE(dat_with_FRA, outcome_col, endog_col, treat_lvl, ctrl_lvl)
# using regression-adjusted Wald-style estimator
# Inputs:
#   dat_with_FRA: dataframe with regression adjusted columns
#   outcome_col: name of outcome whose LATE is being estimated
#   endog_col: treatment, which experiment assignment instruments for
#   treat_lvl: value of W corresponding to "treatment"
#   ctrl_lvl: value of W corresponding to "control"
#
# Output:
#   Vector with point estimate and standard error
#####
FRA_LATE <- function(dat_with_FRA, outcome_col = 'Y', endog_col = 'D', treat_lvl, ctrl_lvl) {
  tmp <- dat_with_FRA %>%
    mutate(u_num = !!sym(paste('u_', outcome_col, '_', treat_lvl, sep = '')) -
             !!sym(paste('u_', outcome_col, '_', ctrl_lvl, sep = '')),
           u_denom = !!sym(paste('u_', endog_col, '_', treat_lvl, sep = '')) -
             !!sym(paste('u_', endog_col, '_', ctrl_lvl, sep = '')))

  pe <- mean(tmp$u_num) / mean(tmp$u_denom)
  VCV <- 1/nrow(dat_with_FRA) * matrix(c(var(tmp$u_num), cov(tmp$u_num, tmp$u_denom),
                                         cov(tmp$u_num, tmp$u_denom), var(tmp$u_denom)), nrow = 2)
  D <- c(1 / mean(tmp$u_denom), - mean(tmp$u_num) / mean(tmp$u_denom)^2)

  c(pe, sqrt(D %*% VCV %*% D))
}
#####


# Estimate function of potential outcome means after regression adjustment
# FRA_theta(para_func, dat_with_FRA, outcome_treats)
# Inputs:
#   param_func: function of potential outcome means being estimated
#   dat_with_FRA: dataframe with regression adjusted columns
#   outcome_treats: vector of strings of the form '{outcome name}_{treatment name}' which
#   are the inputs into param_func
# Output:
#   Vector with point estimate and standard error
#####
FRA_theta <- function(param_func, dat_with_FRA, outcome_treats) {
  input_cols = sapply(outcome_treats, function(x) paste('u_', x, sep = ''))
  VCV = matrix(sapply(input_cols, function(x) sapply(input_cols, function(y)
    cov(dat_with_FRA[,x], dat_with_FRA[,y]))),
    nrow = length(outcome_treats))
  m = as.vector(sapply(input_cols, function(x) mean(dat_with_FRA[,x])))

  D <- grad(param_func, m)
  pe = param_func(m)
  se = sqrt(1/nrow(dat_with_FRA) * D %*% VCV %*% D)
  c(pe, se)
}
#####



# GOTV
#####
data(GerberGreenImai)
dat <- GerberGreenImai
rm(GerberGreenImai)
dat <- dat %>% mutate(Y = VOTED98, W = APPEAL) %>%
  select(Y,W,WARD, AGE, MAJORPTY, VOTE96.0, VOTE96.1, NEW)
dat$WARD <- as.factor(dat$WARD)

set.seed(6124)
covariate_cols <- dat %>% colnames %>% tail(ncol(dat) - 2)
dat_with_FRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = covariate_cols, method = 'rf', n_folds = 10)
dat_with_LRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = covariate_cols, method = 'linear', n_folds = 10)


FRA_ATE(dat_with_FRA, treat_lvl = 3, ctrl_lvl = 1)
FRA_ATE(dat_with_LRA, treat_lvl = 3, ctrl_lvl = 1)

dat %>% group_by(W) %>% summarise(m = mean(Y), v = var(Y) / n()) %>%
  summarise(pe = mean(m[W==3]) - mean(m[W==1]), se = sqrt(mean(v[W==3]) + mean(v[W==1])))
#####

# Ferraro Price
#####
set.seed(326)

# Unlogged Everything
dat <- read_csv('dat_for_RA_ferraroprice.csv') %>% na.omit %>% filter(Y < 200)
dat$Y %>% hist
hist(dat$Y)
covariate_cols <- dat %>% colnames %>% tail(ncol(dat) - 2)

dat_with_FRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = covariate_cols, method = 'gbm', n_folds = 3,
                    num_trees = 600)
dat_with_LRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = covariate_cols, method = 'linear', n_folds = 10)

FRA_ATE(dat_with_FRA, outcome_col = 'Y', treat_lvl = 3, ctrl_lvl = 4)
FRA_ATE(dat_with_LRA, outcome_col = 'Y', treat_lvl = 3, ctrl_lvl = 4)

dat %>% summarise(
  pe = mean(Y[W==3]) - mean(Y[W==4]),
  se = sqrt(var(Y[W==3]) / sum(W==3) + var(Y[W==3])/sum(W==3)))


# Logged Outcome Only
dat <- read_csv('dat_for_RA_ferraroprice.csv') %>% na.omit %>% filter(Y < 200)
dat$Y %>% hist
dat$Y <- log(dat$Y + 1)
hist(dat$Y)
covariate_cols <- dat %>% colnames %>% tail(ncol(dat) - 2)

dat_with_FRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = covariate_cols, method = 'gbm', n_folds = 3,
                    num_trees = 600)
dat_with_LRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = covariate_cols, method = 'linear', n_folds = 10)

FRA_ATE(dat_with_FRA, outcome_col = 'Y', treat_lvl = 3, ctrl_lvl = 4)
FRA_ATE(dat_with_LRA, outcome_col = 'Y', treat_lvl = 3, ctrl_lvl = 4)

dat %>% summarise(
  pe = mean(Y[W==3]) - mean(Y[W==4]),
  se = sqrt(var(Y[W==3]) / sum(W==3) + var(Y[W==3])/sum(W==3)))


# Logged Everything
dat <- read_csv('dat_for_RA_ferraroprice_logged.csv') %>% na.omit %>% filter(Y < 200)
dat$Y %>% hist
dat$Y <- log(dat$Y + 1)
hist(dat$Y)
covariate_cols <- dat %>% colnames %>% tail(ncol(dat) - 2)

dat_with_FRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = covariate_cols, method = 'gbm', n_folds = 3,
                    num_trees = 600)
dat_with_LRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = covariate_cols, method = 'linear', n_folds = 10)

FRA_ATE(dat_with_FRA, outcome_col = 'Y', treat_lvl = 3, ctrl_lvl = 4)
FRA_ATE(dat_with_LRA, outcome_col = 'Y', treat_lvl = 3, ctrl_lvl = 4)

dat %>% summarise(
  pe = mean(Y[W==3]) - mean(Y[W==4]),
  se = sqrt(var(Y[W==3]) / sum(W==3) + var(Y[W==3])/sum(W==3)))
#####

# CHECC
#####
```

```
dat <- read_csv('dat_for_RA_CHECC.csv')
dat$hl <- as.factor(dat$hl)

set.seed(161)
covariate_cols <- dat %>% colnames %>% tail(ncol(dat) - 2)
dat_with_FRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = covariate_cols, method = 'rf', n_folds = 10)
dat_with_LRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = covariate_cols, method = 'linear', n_folds = 10)


dat_with_FRA %>% filter(W == 0) %>%
  summarise(Y_sd_0 = sd(Y), rmse_0 = sqrt(mean((Y-m_Y_0)^2)))
dat_with_FRA %>% filter(W == 1) %>%
  summarise(Y_sd_0 = sd(Y), rmse_0 = sqrt(mean((Y-m_Y_1)^2)))

dat_with_LRA %>% filter(W == 0) %>%
  summarise(Y_sd_0 = sd(Y), rmse_0 = sqrt(mean((Y-m_Y_0)^2)))
dat_with_LRA %>% filter(W == 1) %>%
  summarise(Y_sd_0 = sd(Y), rmse_0 = sqrt(mean((Y-m_Y_1)^2)))


FRA_ATE(dat_with_FRA, treat_lvl = 1, ctrl_lvl = 0)
FRA_ATE(dat_with_LRA, treat_lvl = 1, ctrl_lvl = 0)

dat %>% group_by(W) %>% summarise(m = mean(Y), v = var(Y) / n()) %>%
  summarise(pe = mean(m[W==1]) - mean(m[W==0]),
            se = sqrt(mean(v[W==1]) + mean(v[W==0])))
#####

# OHIE
#####
dat <- read_csv('dat_for_RA_OHIE.csv') %>% na.omit


covariate_cols <- dat %>% colnames %>% tail(ncol(dat) - 3)
set.seed(623)
dat_with_FRA <- FRA(dat, outcome_cols = c('Y','D'),
                    covariate_cols = covariate_cols, method = 'rf', n_folds = 3)
dat_with_LRA <- FRA(dat, outcome_cols = c('Y','D'),
                    covariate_cols = covariate_cols, method = 'linear', n_folds = 10)


dat_with_FRA %>% filter(W == 0) %>%
  summarise(Y_sd_0 = sd(Y), rmse_0 = sqrt(mean((Y-m_Y_0)^2)))
dat_with_FRA %>% filter(W == 1) %>%
  summarise(Y_sd_0 = sd(Y), rmse_0 = sqrt(mean((Y-m_Y_1)^2)))

dat_with_LRA %>% filter(W == 0) %>%
  summarise(Y_sd_0 = sd(Y), rmse_0 = sqrt(mean((Y-m_Y_0)^2)))
dat_with_LRA %>% filter(W == 1) %>%
  summarise(Y_sd_0 = sd(Y), rmse_0 = sqrt(mean((Y-m_Y_1)^2)))



FRA_ATE(dat_with_FRA, treat_lvl = 1, ctrl_lvl = 0)
FRA_ATE(dat_with_LRA, treat_lvl = 1, ctrl_lvl = 0)

(FRA_ATE(dat_with_FRA, treat_lvl = 1, ctrl_lvl = 0)[2]/
    FRA_ATE(dat_with_LRA, treat_lvl = 1, ctrl_lvl = 0)[2])^2


FRA_ATE(dat_with_FRA, outcome_col = 'D', treat_lvl = 1, ctrl_lvl = 0)
FRA_ATE(dat_with_LRA, outcome_col = 'D', treat_lvl = 1, ctrl_lvl = 0)


dat %>% summarise(
  pe_rf = mean(Y[W==1]) - mean(Y[W==0]),
  se_rf = sqrt(var(Y[W==1]) / sum(W==1) + var(Y[W==0])/sum(W==0)),
  pe_fs = mean(D[W==1]) - mean(D[W==0]),
  se_fs = sqrt(var(D[W==1]) / sum(W==1) + var(D[W==0])/sum(W==0)))


FRA_LATE(dat_with_FRA, treat_lvl = 1, ctrl_lvl = 0)
FRA_LATE(dat_with_LRA, treat_lvl = 1, ctrl_lvl = 0)
dat %>% felm(Y~1|0|(D~W), data = .) %>%
  summary


(FRA_LATE(dat_with_FRA, treat_lvl = 1, ctrl_lvl = 0)[2]/
FRA_LATE(dat_with_LRA, treat_lvl = 1, ctrl_lvl = 0)[2])^2

dat %>%
  felm(formula(paste('Y~',paste(covariate_cols,collapse='+'),'|0|(D~W)')),
       data = .) %>%
  summary
#####





# Simulation example
#####
# Latent variables L1, L2, L3


get_pe <- function(p, interaction,N = 1000, method = 'rf') {
  W <- sample(c(0,1), N, replace = T)
  L1 <- runif(N, 0, 1)
  L2 <- runif(N, 0, 1)
  L3 <- runif(N, 0, 1)
  if(interaction == 1) {
    X1 <- (L1 * L2)^p
    X2 <- (L2 * L3)^p
    X3 <- (L3 * L1)^p
  } else {
    X1 <- L1^p
    X2 <- L2^p
    X3 <- L3^p
  }
  U <- rnorm(N, 0, 0.5)

  Y <- L1 + L2 + L3 + W + U

  dat <- data.frame(W = W, X1 = X1, X2 = X2, X3 = X3, Y = Y)



  # Apply regression adjustment pre-processing
  dat_with_FRA <- FRA(dat, outcome_cols = c('Y'), method = method, n_folds = 5)


  # Compare FRA_theta with FRA_ATE estimates of average effect
  FRA_ATE(dat_with_FRA, treat_lvl = 1, ctrl_lvl = 0)[1]
}


set.seed(216)
for (p in c(1, 5,10)) {
  for (interaction in c(0,1)) {
    fits_ml <- sapply(1:100, function(x) get_pe(p,interaction, method = 'rf'))
    fits_linear <- sapply(1:100, function(x) get_pe(p,interaction, method = 'linear'))
    print(c(p,interaction, sd(fits_ml) / sd(fits_linear)))
  }
}


N <- 1000
p <- 1
interaction = 1

W <- sample(c(0,1), N, replace = T)
L1 <- runif(N, 0, 1)
L2 <- runif(N, 0, 1)
L3 <- runif(N, 0, 1)
if(interaction == 1) {
  X1 <- (L1 * L2)^p
  X2 <- (L2 * L3)^p
  X3 <- (L3 * L1)^p
} else {
  X1 <- L1^p
  X2 <- L2^p
  X3 <- L3^p
}
U <- rnorm(N, 0, 0.5)

Y <- L1 + L2 + L3 + W + U

dat <- data.frame(W = W, X1 = X1, X2 = X2, X3 = X3, L1 = L1, L2 = L2, L3 = L3, Y = Y)

# Apply regression adjustment pre-processing
dat_with_FRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = c('X1', 'X2', 'X3'), method = 'rf',n_folds = 5)
dat_with_LRA <- FRA(dat, outcome_cols = c('Y'),
                    covariate_cols = c('X1', 'X2', 'X3'), method = 'linear',n_folds=5)


# Compare FRA_theta with FRA_ATE estimates of average effect
FRA_ATE(dat_with_FRA, treat_lvl = 1, ctrl_lvl = 0)
FRA_ATE(dat_with_LRA, treat_lvl = 1, ctrl_lvl = 0)
#####


# R2 Simulations
#####
get_row <- function(N,p,interaction) {
  W <- sample(c(0,1), N, replace = T)
  L1 <- runif(N, 0, 1)
  L2 <- runif(N, 0, 1)
  L3 <- runif(N, 0, 1)
  if(interaction == 1) {
    X1 <- (L1 * L2)^p
    X2 <- (L2 * L3)^p
    X3 <- (L3 * L1)^p
  } else {
    X1 <- L1^p
    X2 <- L2^p
    X3 <- L3^p
  }
  U <- rnorm(N, 0, 0.5)

  Y <- L1 + L2 + L3 + W + U

  dat <- data.frame(W = W, X1 = X1, X2 = X2, X3 = X3, L1 = L1, L2 = L2, L3 = L3, Y = Y)

  # Apply regression adjustment pre-processing
  dat_with_FRA <- FRA(dat, outcome_cols = c('Y'),
                      covariate_cols = c('X1', 'X2', 'X3'), method = 'rf',n_folds = 5)
  dat_with_LRA <- FRA(dat, outcome_cols = c('Y'),
                      covariate_cols = c('X1', 'X2', 'X3'), method = 'linear',n_folds=5)


  FRA_UV <- dat_with_FRA %>% group_by(W) %>%
    summarise(var_tot = var(Y), var1 = var(Y - m_Y_1), var0 = var(Y - m_Y_0), n = n()) %>%
    summarise(var_tot = sum(n*var_tot)/sum(n),
              var_reduced = (sum(n[W==0]*var0[W==0]) + sum(n[W==1]*var0[W==1])) /
                (sum(n[W==0]) + sum(n[W==1]))) %>%
    mutate(UV = var_reduced / var_tot) %>% .$UV

  LRA_UV <- dat_with_LRA %>% group_by(W) %>%
    summarise(var_tot = var(Y), var1 = var(Y - m_Y_1), var0 = var(Y - m_Y_0), n = n()) %>%
    summarise(var_tot = sum(n*var_tot)/sum(n),
              var_reduced = (sum(n[W==0]*var0[W==0]) + sum(n[W==1]*var0[W==1])) /
                (sum(n[W==0]) + sum(n[W==1]))) %>%
    mutate(UV = var_reduced / var_tot) %>% .$UV

  var_SDM <- var(dat_with_FRA$Y[dat_with_FRA$W==1]) / nrow(dat_with_FRA[dat_with_FRA$W==1,]) +
    var(dat_with_FRA$Y[dat_with_FRA$W==0]) / nrow(dat_with_FRA[dat_with_FRA$W==0,])

  # Compare FRA_theta with FRA_ATE estimates of average effect
  var_FRA <- FRA_ATE(dat_with_FRA, treat_lvl = 1, ctrl_lvl = 0)[2]^2
  var_LRA <- FRA_ATE(dat_with_LRA, treat_lvl = 1, ctrl_lvl = 0)[2]^2
  #####
  data.frame(FRA_UV = FRA_UV, LRA_UV = LRA_UV, FRA_VR = var_FRA / var_SDM,
             LRA_VR = var_LRA / var_SDM)
}


set.seed(10311)
plot_dat <- NULL
for (N in c(30, 60, 90, 120, 150)) {
  for(p in c(1,5,10)) {
    row <- NULL
    for (i in 1:100) {
      row <- rbind(row, get_row(N,p,T))
    }
    row <- row %>% summarise(FRA_UV = mean(FRA_UV), LRA_UV = mean(LRA_UV),
                             FRA_VR = mean(FRA_VR), LRA_VR = mean(LRA_VR))
    row$N <- N
    row$p <- p
    plot_dat <- plot_dat %>% rbind(row)
  }
}


plot_dat %>% ggplot + geom_point(aes(x = FRA_UV, y = FRA_VR, col = as.factor(p), size = N)) +
  geom_abline(aes(intercept=0,slope=1), size = 2) +
  labs(x = expression('1-R'^'2'), y = 'Variance Shrinkage Factor') +
  scale_color_discrete(name = 'p') + theme(axis.text=element_text(size=16),
                                           axis.title=element_text(size=18),
                                           legend.title=element_text(size=15),
                                           legend.text=element_text(size=14))
#####
```

C Tables

Table 1: Summmary Statistics of Normalized Treatment Effect Estimates

Simulation	Mean	Std. Dev.	95% CI Coverage	99% CI Coverage
TE, Moderate, Cross	-0.012	1.00	0.9521	0.9887
TE, Moderate, No Cross	-1.33	1.06	0.8792	0.7256
Null, Moderate, Cross	0.005	1.00	0.9509	0.9904
Null, Moderate, No Cross	-0.001	0.99	0.9507	0.9906
TE, Large, Cross	0.017	1.02	0.946	0.989
TE, Large, No Cross	-1.16	1.07	0.764	0.894
Null, Large, Cross	-0.04	0.96	0.959	0.990
Null, Large, No Cross	-0.02	1.03	0.947	0.989
TE, Small, Cross	-0.019	1.00	0.95191	0.99181
TE, Small, No Cross	-1.06	1.10	0.91515	0.78837
Null, Small, Cross	0.004	1.00	0.95264	0.99242
Null, Small, No Cross	-0.001	1.01	0.9495	0.9885
Ratio, Moderate, Cross	-0.01	1.00	0.95264	0.99242
Ratio, Moderate, No Cross	-0.005	1.20	0.8983	0.9675

Notes: This table contains summary statistics for the normalized treatment value estimates. If the asymptotic theory is valid, these normalized estimates should be mean zero with a standard deviation of 1, and the 95% and 99% CIs should cover 95% and 99% of the values respectively. Each simulation is indexed by (TE, Sample Size, Cross) where TE and Cross denote respectively whether or not the dataset has a treatment effect and whether or not cross fitting was used. We refer to the simulations obtained by splitting the original dataset into 1,000, 10,000, and 100,000 pieces respectively as “Large”, “Moderate”, and “Small” samples. The “Large” sample had a number of observations in the tens of thousands while the “Small” sample had a number of observations in the hundreds. The last two rows contain results from estimating a ratio metric.

Table 2: Shrinkage Factor for Confidence Intervals

Simulation	Ratio
Small, Cross	0.85
Small, No Cross	0.24
Moderate, Cross	0.75
Moderate, No Cross	0.27
Large, Cross	0.67
Large, No Cross	0.38
Ratio, Cross	0.96
Ratio, No Cross	0.80

Notes: This table shows the factor by which the estimated confidence intervals shrink depending on whether or not sample splitting is used, what data is being used, and whether an average treatment effect or a difference in ratios is being estimated.

Table 3: Variance Reduction for Average Treatment Effects

	$p\!=\!1$	$p\!=\!5$	$p\!=\!10$
Interaction	1.03	0.94	0.72
No Interaction	0.93	0.81	0.68

Notes: This table displays the ratio between the standard deviation of the FRA estimator relative to the standard deviation of the LRA estimator. The columns vary non-linearity, as parameterized by p while the rows vary whether or not there are interactions between covariates.

Table 4: Variance Reduction for Average Treatment Effects

	SM	LRA	FRA
Cancel Rate	-0.60%	-1.26%	-1.32%
	(0.52)	(0.35)	(0.34)
No Show Rate	-4.4%	-4.2%	-4.3%
	(0.51)	(0.40)	(0.40)

Notes: This table shows point estimates and standard errors in parentheses for the percent difference in cancel rate and no show rate between treatment and control. The first column looks at the estimator obtained by plugging in the subsample means. The second column considers a linear regression adjustment. The third column uses a nonparamteric regression adjustment. We are unable to report sample sizes for this analysis.

Table 5: R^2 in predicting outcomes

	# Accepts	# Cancels	# No Shows
Linear R^2	0.687	0.587	0.514
Flexible R^2	0.712	0.580	0.525

Notes: This table shows the R^2 in predicting various metrics necessary to compute cancellation and no show rates. The first row considers R^2 from fitting an OLS model while the second row considers the R^2 from fitting a flexible non-parametric model.

Table 6: Reduction in Standard Errors

	# Accepts	# Cancels	# No Shows
Diff-in-diff	0.622	0.667	0.786
Regression Adjustment	0.596	0.658	0.721

Notes: This table shows the reduction in standard errors from using different variance reduction techniques. The columns index the outcome measure while the rows index the estimator considered. For a given outcome Y and for a given estimator $\hat{\beta}$, each entry in the table displays the ratio between the standard errors from fitting $\hat{\beta}$ on outcome Y relative to the standard errors from taking a simple difference in means.

Table 7: Variance Reduction for OHIE

	SM	LRA	FRA
ER Visits	0.0132	0.0143	0.0139
	(0.0085)	(0.0079)	(0.0077)
Medicaid Take-Up	0.172	0.159	0.150
	(0.0063)	(0.0062)	(0.0062)
LATE	0.0892	0.0902	0.0870
	(0.0496)	(0.0498)	(0.0482)

Notes: This table shows point estimates and standard errors in parentheses for a number of causal parameters from the OHIE across a number of regression adjustment specifications. In the first row, we measure the reduced form impact of treatment assignment on ER visits. In the second row, we measure the first stage impact of treatment assignment on Medicaid take-up. In the third row, we divide the first row by the second row to obtain an estimate of the LATE of Medicaid uptake. The sample size is $N = 13,051$.

Table 8: Variance Reduction for Water Conservation

	SM	LRA	FRA
Un-Logged Outcomes and Covariates	-1.44	-1.84	-1.85
	(0.354)	(0.160)	(0.155)
Logged Outcomes, Unlogged Covariates	-0.0293	-0.0365	-0.0368
	(0.00860)	(0.00463)	(0.00402)
Logged Outcomes and Covariates	-0.0293	-0.0374	-0.0377
	(0.00860)	(0.00415)	(0.00406)

Notes: This table shows point estimates and standard errors in parentheses for a number of causal parameters from the OHIE across a number of regression adjustment specifications. In the first row, we measure raw outcomes and take raw pre-exposure outcomes as covariates. In the second row, we measure log outcomes. In the third row, we also apply the log transformation to covariates. The sample size is $N = 100,026$

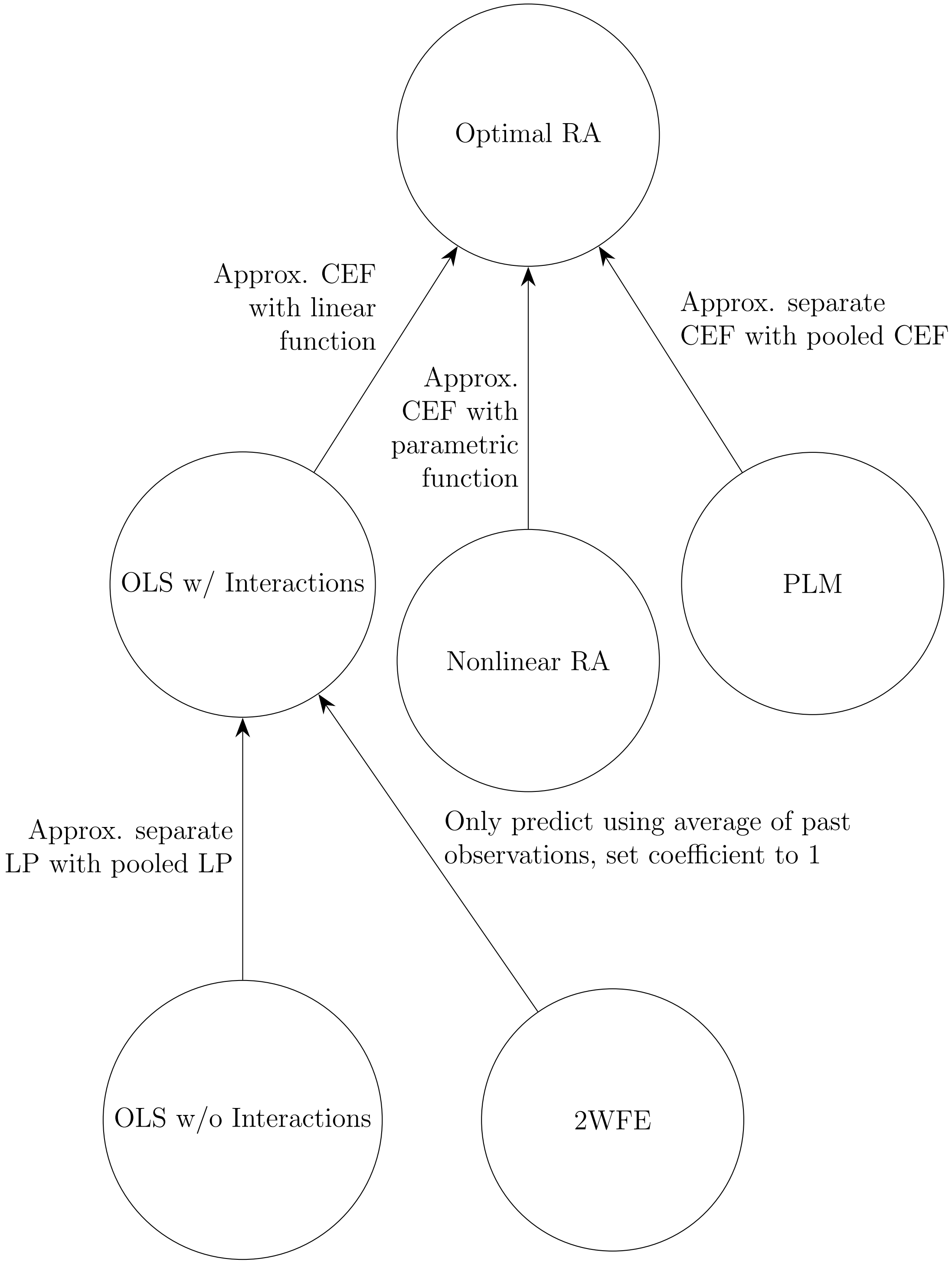
Table 9: Variance Reduction for CogX

	SM	LRA	FRA
Cog Test Scores	7.13	10.75	8.97
	(2.69)	(2.59)	(2.49)

Notes: This table shows point estimates and standard errors in parentheses for the effect of the CogX program on cognitive test scores across a number of regression adjustment specifications. The sample size is $N = 395$.

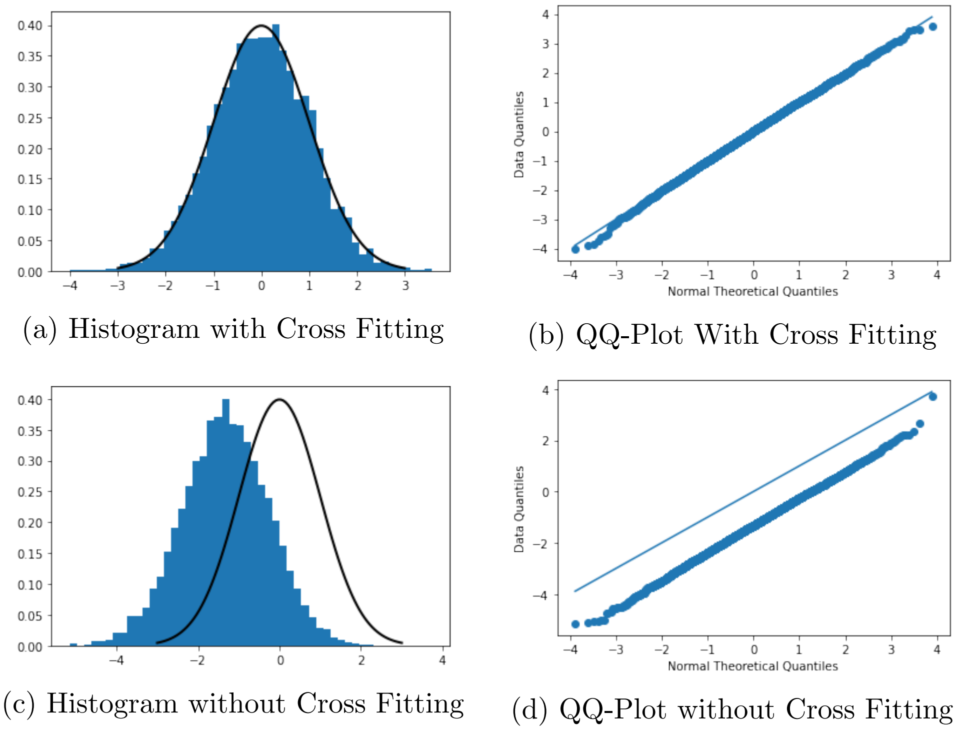
D Figures

Figure 1: Comparison of Variance Reduction Strategies


Optimal RA
Approx. CEF with linear function
Approx. CEF with parametric function
Approx. separate CEF with pooled CEF
OLS w/ Interactions
Nonlinear RA
PLM
Approx. separate LP with pooled LP
Only predict using average of past observations, set coefficient to 1
OLS w/o Interactions
2WFE

Notes: This figure summarizes the comparison of the optimal regression adjustment derived in this paper with common alternative variance reduction strategies. Nodes in the graph represent estimators. Each edge points towards an estimator which is unambiguously lower variance than the edge being pointed away from, and edge text provides intuition for where the inefficiency stems from. The meaning of abbreviations are as follows: RA = regression adjustment, CEF = conditional expectation function, PLM = partially linear model, LP = linear predictor, 2WFE = two-way fixed effect.

Figure 2: Distribution of Normalized Estimates (TE, Moderate)



(a) Histogram with Cross Fitting

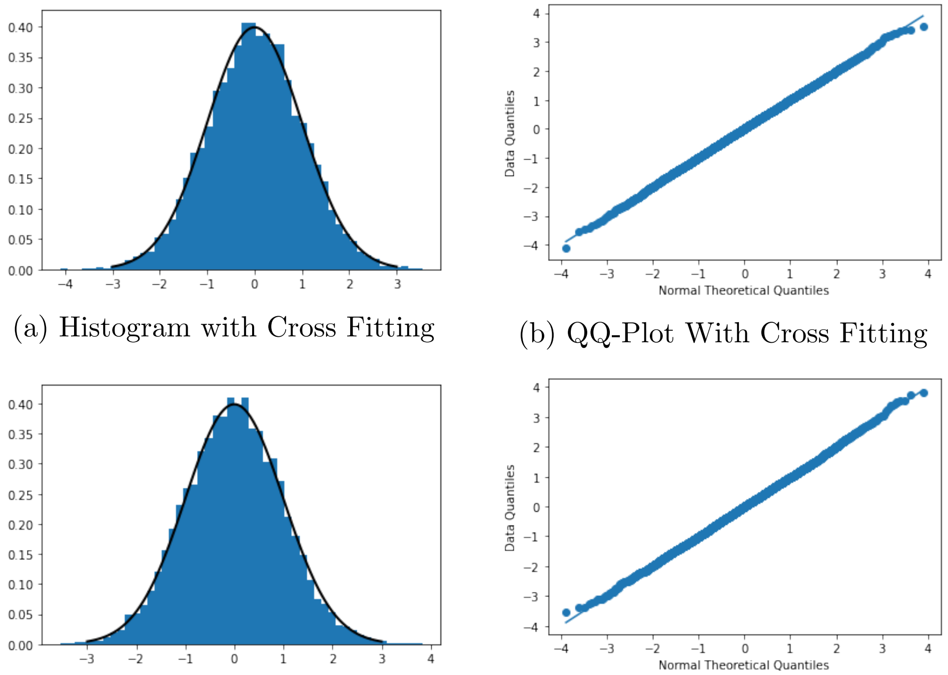
(b) QQ-Plot With Cross Fitting

(c) Histogram without Cross Fitting

(d) QQ-Plot without Cross Fitting

Notes: This figure plots the distribution of normalized estimates of the treatment effect, which subtract off the population average treatment effect and divide by the sample standard deviation. Panels (a) and (b) show the distribution from a procedure where sample splitting is used while Panels (c) and (d) show the distribution from a procedure where sample splitting is not used.

Figure 3: Distribution of Normalized Estimates (Null, Moderate)



(a) Histogram with Cross Fitting

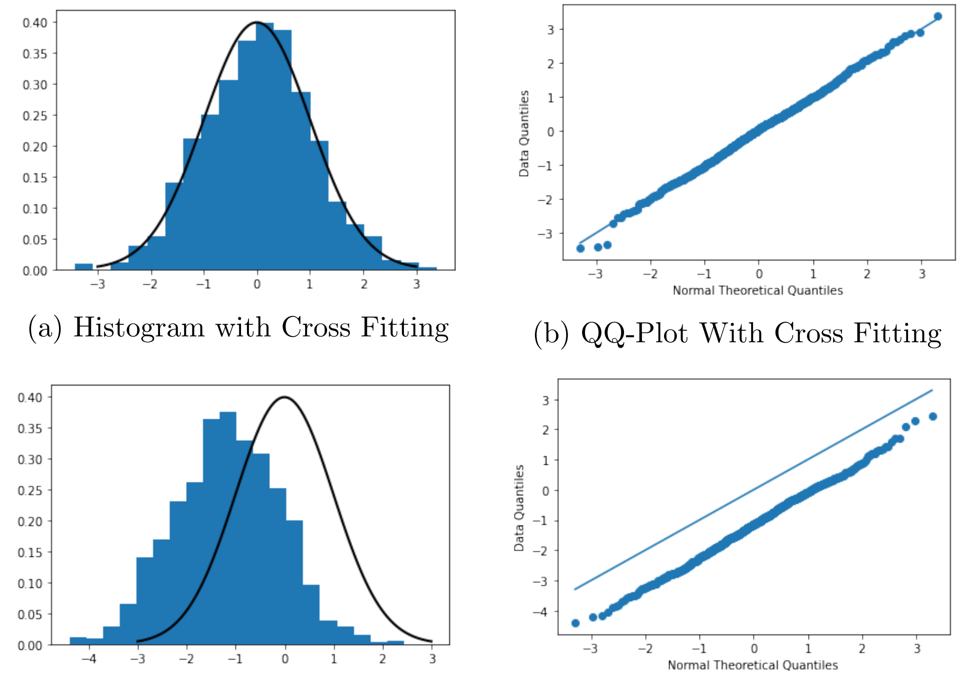
(b) QQ-Plot With Cross Fitting

(c) Histogram without Cross Fitting

(d) QQ-Plot without Cross Fitting

Notes: This figure plots the distribution of normalized estimates of the treatment effect, which subtract off the population average treatment effect and divide by the sample standard deviation. The dataset being used here by construction has no treatment effects. Panels (a) and (b) show the distribution from a procedure where sample splitting is used while Panels (c) and (d) show the distribution from a procedure where sample splitting is not used.

Figure 4: Distribution of Normalized Estimates (TE, Large)



(a) Histogram with Cross Fitting

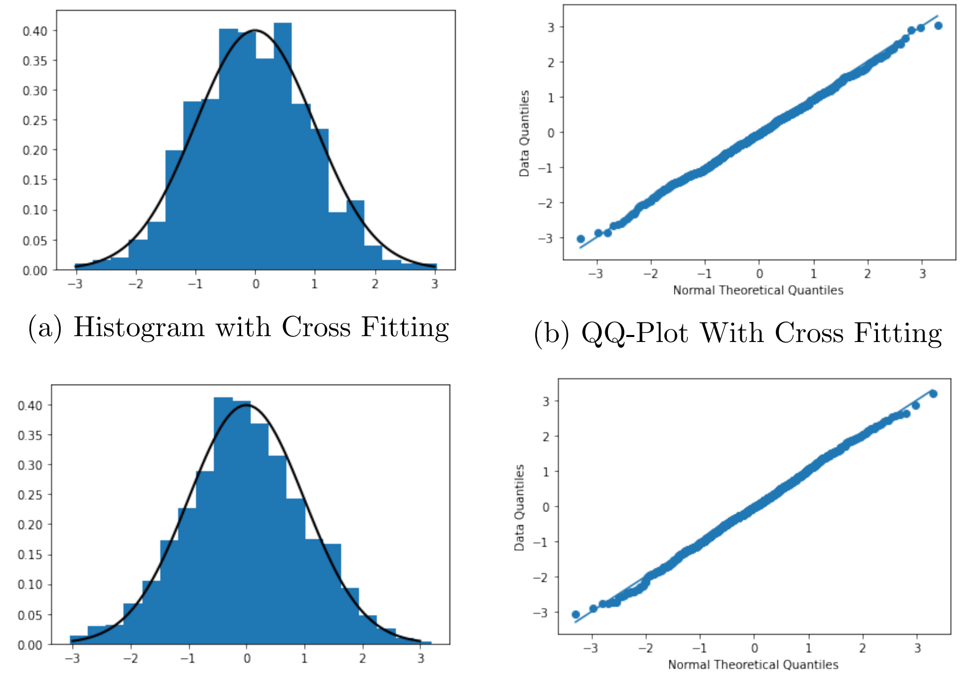
(b) QQ-Plot With Cross Fitting

(c) Histogram without Cross Fitting

(d) QQ-Plot without Cross Fitting

Notes: This figure plots the distribution of normalized estimates of the treatment effect, which subtract off the population average treatment effect and divide by the sample standard deviation. Compared to Figure 2, we use sample sizes that are 10 times as large. Panels (a) and (b) show the distribution from a procedure where sample splitting is used while Panels (c) and (d) show the distribution from a procedure where sample splitting is not used.

Figure 5: Distribution of Normalized Estimates (Null, Large)



(a) Histogram with Cross Fitting

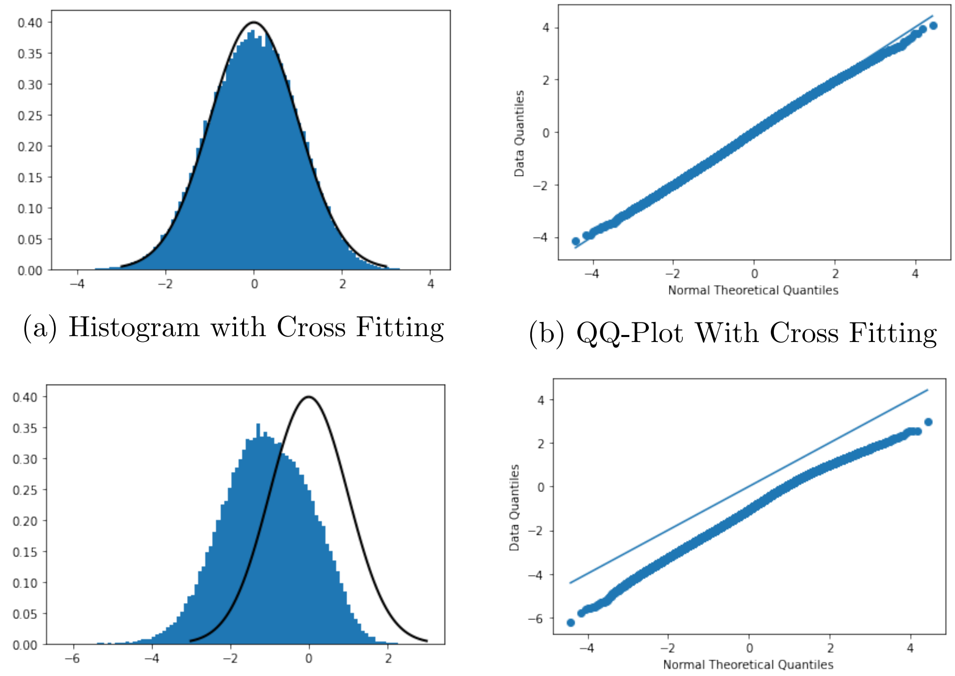
(b) QQ-Plot With Cross Fitting

(c) Histogram without Cross Fitting

(d) QQ-Plot without Cross Fitting

Notes: This figure plots the distribution of normalized estimates of the treatment effect, which subtract off the population average treatment effect and divide by the sample standard deviation. The dataset being used here by construction has no treatment effects. Compared to Figure 2, we use sample sizes that are 10 times as large. Panels (a) and (b) show the distribution from a procedure where sample splitting is used while Panels (c) and (d) show the distribution from a procedure where sample splitting is not used.

Figure 6: Distribution of Normalized Estimates (TE, Small)



(a) Histogram with Cross Fitting

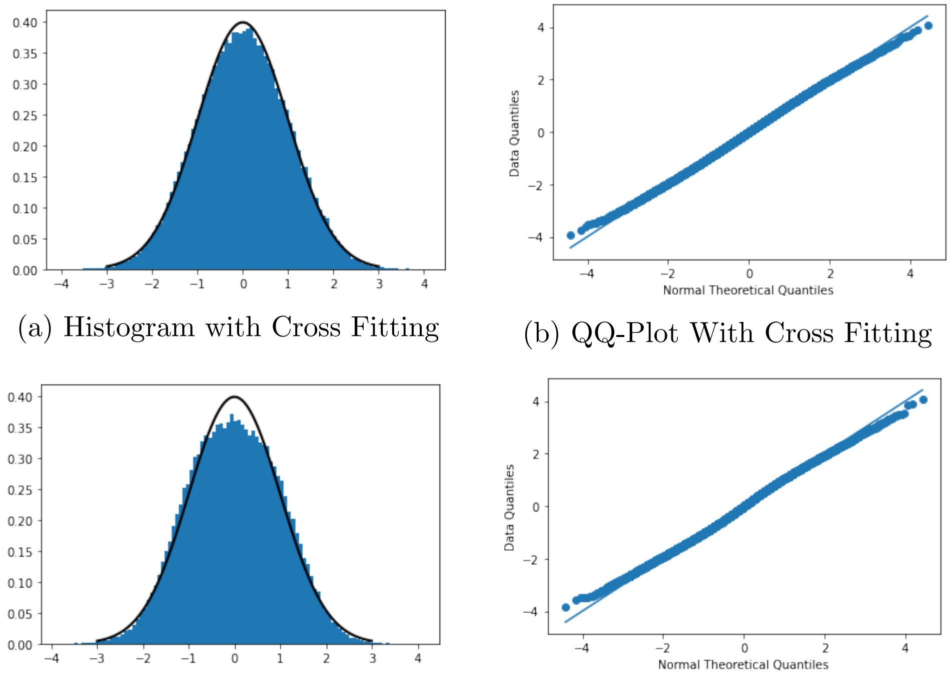
(b) QQ-Plot With Cross Fitting

(c) Histogram without Cross Fitting

(d) QQ-Plot without Cross Fitting

Notes: This figure plots the distribution of normalized estimates of the treatment effect, which subtract off the population average treatment effect and divide by the sample standard deviation. Compared to Figure 2, we use sample sizes that are 10 times as small. Panels (a) and (b) show the distribution from a procedure where sample splitting is used while Panels (c) and (d) show the distribution from a procedure where sample splitting is not used.

Figure 7: Distribution of Normalized Estimates (Null, Small)



(a) Histogram with Cross Fitting

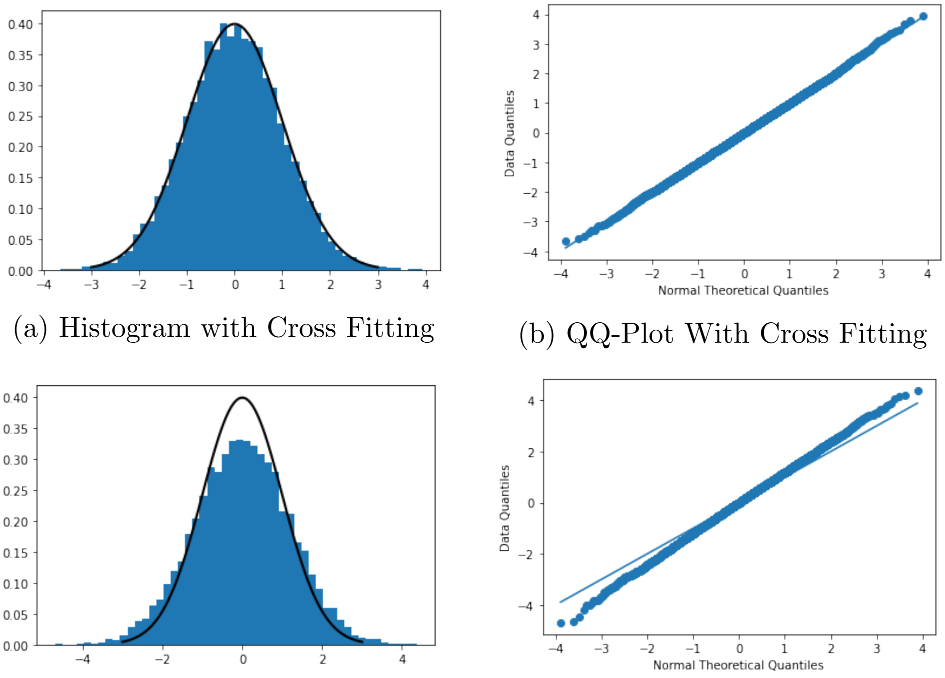
(b) QQ-Plot With Cross Fitting

(c) Histogram without Cross Fitting

(d) QQ-Plot without Cross Fitting

Notes: This figure plots the distribution of normalized estimates of the treatment effect, which subtract off the population average treatment effect and divide by the sample standard deviation. The dataset being used here by construction has no treatment effects. Compared to Figure 3, we use sample sizes that are 10 times as small. Panels (a) and (b) show the distribution from a procedure where sample splitting is used while Panels (c) and (d) show the distribution from a procedure where sample splitting is not used.

Figure 8: Distribution of Normalized Estimates (Ratio)



(a) Histogram with Cross Fitting

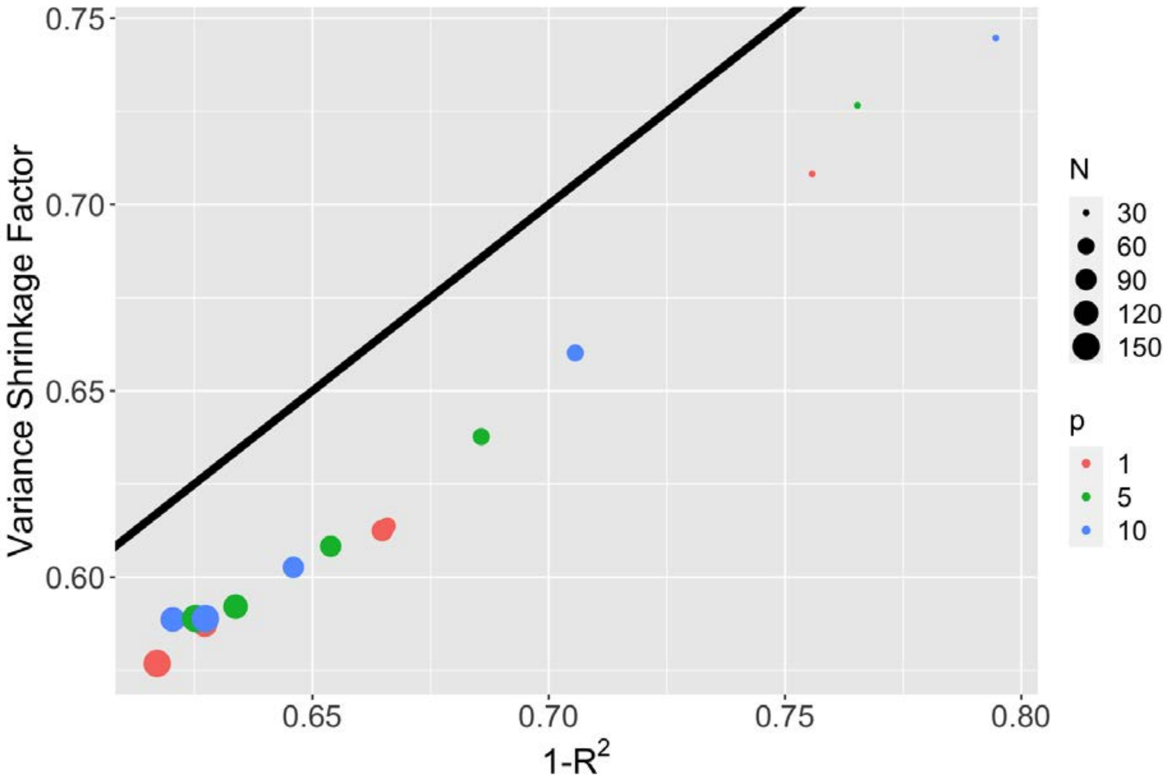
(b) QQ-Plot With Cross Fitting

(c) Histogram without Cross Fitting

(d) QQ-Plot without Cross Fitting

Notes: This figure plots the variance reduction from regression adjustment ($\frac{\hat{\sigma}_f^2}{\hat{\sigma}_s^2}$) on the vertical axis against $1-\hat{R}^2$ from the predictive model underlying the regression adjustment on the horizontal axis. Colors represent different values of p in the data-generating process, while within color, different observations reflect different sample sizes. Sizes of the dots represent the sample size N of the the simulated dataset. The solid black line is the 45° line.

Figure 9: Variance Reduction for Average Treatment Effects



Notes: This figure plots the distribution of normalized estimates of the difference in ratios, which subtract off the population average difference and divide by the sample standard deviation. Panels (a) and (b) show the distribution from a procedure where sample splitting is used while Panels (c) and (d) show the distribution from a procedure where sample splitting is not used.