A Proofs

Proof of Proposition 1 [Anava and Levy, 2016].

$$\begin{aligned}
&\left|\sum_{i=1}^{n} w_{\star,i} y_i - f(x_\star)\right| \\
&= \left|\sum_{i=1}^{n} \alpha_i \left(y_i - f(x_i) + f(x_i)\right) - f(x_0)\right| \\
&= \left|\sum_{i=1}^{n} w_{\star,i}\epsilon_i + \sum_{i=1}^{n} w_{\star,i}\left(f(x_i) - f(x_0)\right)\right| \\
&\leq \left|\sum_{i=1}^{n} w_{\star,i}\epsilon_i\right| + \left|\sum_{i=1}^{n} w_{\star,i}\left(f(x_i) - f(x_\star)\right)\right| \\
&\leq \underbrace{\left|\sum_{i=1}^{n} w_{\star,i}\epsilon_i\right|}_{\text{variance}} + \underbrace{L\sum_{i=1}^{n} w_{\star,i} d(x_i, x_\star)}_{\text{bias}}
\end{aligned} \tag{6}$$

Where L is the Lipschitz constant as in assumption 3, and d is a distance measure. By placing an additional assumption that the noise term can be bounded by a constant, $|\epsilon| \leq b$, the variance term can be further bounded with probability $1-\delta$ using an application of Hoeffding's inequality [Anava and Levy, 2016],

$$\left|\sum_{i=1}^{n} w_{\star,i}\epsilon_i\right| \leq C\|\mathbf{w}_\star\|_2, C = b\sqrt{2\log(\frac{2}{\delta})}$$

□

Proof of Proposition 2. By triangle inequality, we have for every i,

$$\left|\frac{\sum_j \mathbb{1}(a_i \neq a_j)e_{ij}y_j}{\sum_k \mathbb{1}(a_i \neq a_k)e_{ik}} - y_i\right| \leq \left|\frac{\sum_j \mathbb{1}(a_i \neq a_j)e_{ij}y_j}{\sum_k \mathbb{1}(a_i \neq a_k)e_{ik}} - \frac{\sum_j e_{ij}y_j}{\sum_k e_{ik}}\right| + \left|y_i - \frac{\sum_j e_{ij}y_j}{\sum_k e_{ik}}\right|.$$

Let us also denote $d_i = \sum_j e_{ij}$ and $d_i(\text{cut}) = \sum_j \mathbb{1}(a_i \neq a_j)e_{ij}$.

$$\begin{aligned}
\left|\sum_j \frac{e_{ij}y_j}{\sum_k e_{ik}} - \sum_j \frac{\mathbb{1}(a_i \neq a_j)e_{ij}y_j}{\sum_k \mathbb{1}(a_i \neq a_j)e_{ik}}\right| &= \left|\sum_j \frac{e_{ij}y_j}{d_i} - \sum_j \mathbb{1}(a_i \neq a_j)\frac{e_{ij}y_j}{d_i(\text{cut})}\right| \\
&= \left|\sum_j \mathbb{1}(a_i \neq a_j)e_{ij}y_j\left(\frac{1}{d_i} - \frac{1}{d_i(\text{cut})}\right) + \sum_j \frac{(1-\mathbb{1}(a_i \neq a_j))e_{ij}y_j}{d_i}\right| \\
&\leq \left|0 + \sum_j \frac{1-\mathbb{1}(a_i \neq a_j))e_{ij}}{d_i}\right| \\
&= \sum_j \frac{1-\mathbb{1}(a_i \neq a_j))e_{ij}}{d_i} \\
&\leq \frac{d(i) - \sum_j \mathbb{1}(a_i \neq a_j)e_{ij}}{d_{\min}}.
\end{aligned}$$

The first inequality follows by assuming that $0 \leq y_i \leq 1$ and noting that $1/d - 1/d(C) \leq 0$. So the first summand is maximized by setting $y_j = 0$ for $j \in C$ and second one is maximized by setting $y_j = 1$ for $j \in D$. The second inequality follows by $d \geq d_{min}$. Finally, by summing over all i, we have

$$\sum_i \left|\frac{\sum_j \mathbb{1}(a_i \neq a_j)e_{ij}y_j}{\sum_k \mathbb{1}(a_i \neq a_k)e_{ik}} - y_i\right| \leq \sum_i \left|y_i - \frac{\sum_j e_{ij}y_j}{\sum_k e_{ik}}\right| + \frac{e_{\text{sum}} - \sum_{i,j}\mathbb{1}(a_i \neq a_j)e_{ij}}{d_{\min}}.$$

This is minimized by the max-cut. □

A.1 Kallus [2018] and Maxcut

Given the graph, an equivalent formulation of Equation 1 is finding the weighted maximum cut on the graph (find a subset of the vertices such that the total weight of edges connecting nodes of the two different subsets is maximized). The equivalence is made plain by considering the binary quadratic program formulation of Maxcut given in Equation 5,

$$\begin{aligned}&\arg\max_{\mathbf{u}\in\{-1,1\}} \mathbf{u}^T L\mathbf{u} = \arg\max_{\mathbf{u}\in\{-1,1\}} \mathbf{u}^T D\mathbf{u} - \mathbf{u}^T G\mathbf{u}\\ &= \arg\max_{\mathbf{u}\in\{-1,1\}} -\mathbf{u}^T G\mathbf{u}\end{aligned} \tag{7}$$

where we have defined L to be the combinatorial graph Laplacian, $D - G$ where D is a diagonal matrix where D_i is the degree of vertex i and G is the weighted adjacency matrix. Comparing this to problem 1,

$$\begin{aligned}&\arg\min_{\mathbf{u}\in\{-1,1\}} \mathbf{u}^T K\mathbf{u} = \arg\min_{\mathbf{u}\in\{-1,1\}} \mathbf{u}^T \text{diag}(K)\mathbf{u} + \mathbf{u}^T G\mathbf{u}\\ &= \arg\min_{\mathbf{u}\in\{-1,1\}} \mathbf{u}^T G\mathbf{u}\end{aligned}$$

we can see that problem 7 given by maxcut is isomorphic to problem 1, given by Kallus [2017]'s PSOD strategy. Therefore, improved approximations to Maxcut will additionally be improved approximations to Kallus [2018].

A.2 The kernel objective of Kallus [2018] as uncentered Maximum Mean Discrepancy

The kernel objective of Kallus [2018] is defined as

$$\min_{\mathbf{u}\in\{-1,1\}} \mathbf{u}^t K\mathbf{u} \tag{8}$$

where K is the Gram matrix for some reproducing kernel. By using the cyclic properties of the trace, we can rewrite the objective in equation 8 is equivalent to

$$\min_{\mathbf{u}\in\{-1,1\}} \text{trace}\left(K\mathbf{u}\mathbf{u}^t\right) \tag{9}$$

a biased estimator of the Hilbert-Schmidt independence criterion with respect to K and the kernel given by $\mathbf{u}\mathbf{u}^T$ can be written as [Gretton et al., 2008]

$$\begin{aligned}&\text{trace}(KH\mathbf{u}\mathbf{u}^T H)\\ &H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\end{aligned} \tag{10}$$

Equation 10, in turn, was shown to be equivalent to the biased estimator of the Maximum Mean Discrepancy by Song [2008].

B Simulated data generating processes

DGP	$\mathbf{X}$	$y(0)$	$y(1)$
LinearDGP	$X_k = \epsilon_k, k \in \{1,\dots,4\}$	$\mathbf{X}\beta + \frac{1}{10}\epsilon_{y(0)}$	$1 + \mathbf{X}\beta + \frac{1}{10}\epsilon_{y(1)}$
QuickBlockDGP	$X_k \sim \mathcal{U}(0,10), \forall k \in \{1,2\}$	$\prod_{k=1}^{2} X_k + \epsilon$	$1 + \text{y}(0)$
SinusoidalDGP	$X_k = \epsilon_k, k \in \{1,\dots,4\}$	$\sin(\mathbf{X}\beta) + \frac{\epsilon_{y(0)}}{10}$	$1 + \sin(\mathbf{X}\beta) + \frac{\epsilon_{y(1)}}{10}$
TwoCircles	latent: $r \sim \mathcal{N}(1 + i\%2, \frac{1}{10})$ $s \sim \mathcal{U}(0, 2\pi)$ observed: $x_1 = r\cos(s)$, $x_2 = r\sin(s)$	$\beta_1 s + \beta_2 r + \epsilon_{y(0)}$	$\beta_1 s + \beta_2 r + \epsilon_{y(1)}$

Table 1: Data generating processes used in simulations. All ϵs indicate a standard normal variate and all βs indicate a standard uniform variate. % indicates the modulo function, and i indicates a unit's index.

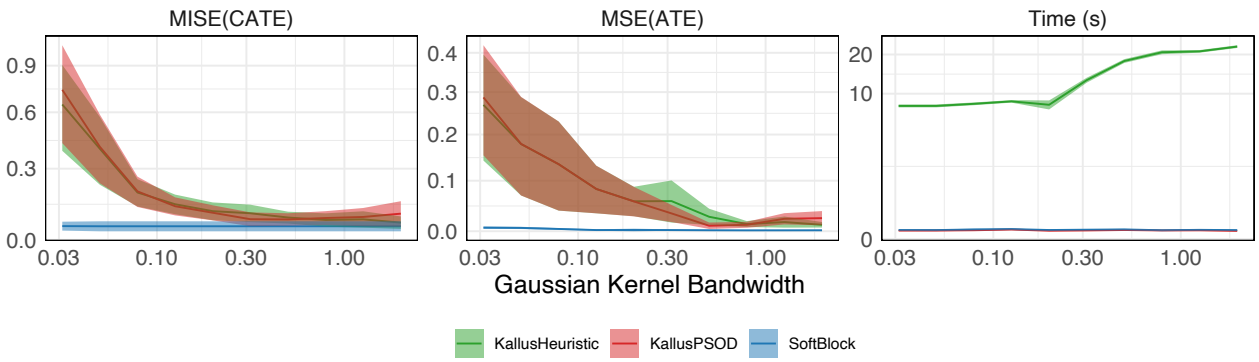

Figure 6: SoftBlock is robust to hyperparameter selection. This figure shows mean squared-error of the ATE, mean integrated-squared-error of the CATE and time to calculate a design and perform estimation.

C Estimators for Experiments

For the estimation of ATEs for Bernoulli and rerandomization, we use regression-adjusted estimators as used in Lin [2013]: a linear regression with covariates mean-centered and interacted with treatment. QuickBlock uses a blocking estimator as the authors propose, and the Kallus [2018] designs use a difference-in-means estimator as proposed. The matched-pairs design takes the average in within-pair outcomes, which leads to a more efficient estimator than difference-in-means [Imai, 2008]. When examining CATE estimators, we use random forest based T-learners unless otherwise noted [Athey and Imbens, 2016, Künzel et al., 2019]. All methods use the same hyperparameters, with the number of trees set at $20 \times n^{\frac{1}{4}}$ to ensure model complexity grows with sample size and with maximum tree depth set at 8.

D Sensitivity to Hyperparameters

Figure 6 shows the sensitivity of the Kallus [2018] methods and SoftBlock to hyperparameters. Since both methods are based on similarities defined by a kernel matrix, we plot the performance of these methods on the TwoCircles problem as the bandwidth of the Gaussian kernel changes. Softblock is not at all sensitive to hyperparameters, performing well at all values, while the Kallus [2018] methods perform well only when the hyper-parameters are set well. In essence, these methods perform covariate adjustment a-priori, but this means that they implicitly specify an outcome model *before data is observed*. As such, it is very difficult to set these values effectively in practice, as it amounts to tuning a non-parametric model without data for cross-validation or other model selection techniques.