Constrained Learning for Causal Inference

Tiffany (Tianhui) Cai*† Yuri Fonseca*‡ Kaiwen Hou+ Hongseok Namkoong†
†Columbia University, ‡Stanford University, +UC Berkeley
tiffany.cai@columbia.edu, yrf2001@stanford.edu, kaiwen.hou@berkeley.edu, namkoong@gsb.columbia.edu

Abstract

Popular debiased estimation methods for causal inference—such as augmented inverse propensity weighting and targeted maximum likelihood estimation—enjoy desirable asymptotic properties like statistical efficiency and double robustness but they can produce unstable estimates when there is limited overlap between treatment and control, requiring additional assumptions or ad hoc adjustments in practice (e.g., truncating propensity scores). In contrast, simple plug-in estimators are stable but lack desirable asymptotic properties. We propose a novel debiasing approach that achieves the best of both worlds, producing stable plug-in estimates with desirable asymptotic properties. Our constrained learning framework solves for the best plug-in estimator under the *constraint* that the first-order error with respect to the plugged-in quantity is zero, and can leverage flexible model classes including neural networks and tree ensembles. In several experimental settings, including ones in which we handle text-based covariates by fine-tuning language models, our constrained learning-based estimator outperforms basic versions of one-step estimation and targeting in challenging settings with limited overlap between treatment and control, and performs similarly otherwise.

1 Introduction

Causal inference is the bedrock of scientific decision-making. In many settings, estimating causal effects accurately requires modeling high-dimensional and complex nuisance parameters—quantities that, while not of primary interest, must be estimated correctly to determine the causal effect. For example, when estimating the average treatment effect (ATE), a common approach is to train a flexible machine learning model to directly model the outcome variable as a function of the treatment assignment and other observed covariates. Then, to estimate the ATE, the trained machine learning model (nuisance parameter) is used to estimate outcomes for each unit under the treatment and under the control. The

ATE is then computed by taking the average of the differences in the predicted values for each observation. This procedure is what we refer to as the naive plug-in (a.k.a. "direct") estimator, which entirely trusts the fitted ML model. Although straightforward and easy to communicate, such an approach has the drawback that the machine learning model is fitted to optimize prediction accuracy and does not consider the downstream causal estimation task. As a result, this naive plug-in estimator is suboptimal and sensitive to errors in the ML model [8, 13, 57, 32].

To improve upon the naive plug-in (direct) estimator, debiased methods analyze the sensitivity of the causal estimand with respect to the estimated nuisance parameter (e.g. outcome model) by taking a first-order distributional Taylor expansion, and then correct for the first-order error term due to nuisance parameter estimation [8, 37, 13, 57, 52, 35]. As an example, *one-step estimation* corrects the plug-in estimator by subtracting an estimate of the first-order error term; see [32, 21] for a review and an introduction to one-step correction. As another example, targeting "fluctuates" the outcome model in a specific way so that an estimate of the first-order error is zero [57, 52]. When estimating the ATE, these two approaches give rise to the well-known augmented inverse probability weighting (AIPW) estimator [5] and the targeted maximum likelihood estimator (TMLE) [57, 52], respectively. Both estimators are statistically efficient/optimal (having the lowest possible asymptotic variance, in the local asymptotic minimax sense [59]) and doubly robust (converging to the true value if either the outcome model or treatment model converge to their true values [5]).

While all first-order correction ("debiased") methods enjoy standard asymptotic optimality guarantees under standard assumptions, in practice, several authors have noted a salient gap between the asymptotic and finite-sample performance of different estimation approaches [31, 11], leaving room for new methodological development, and necessitating a rigorous and thorough empirical comparison. AIPW and TMLE are asymptotically optimal but can produce unstable estimates, and additional heuristics and assumptions (e.g., truncating propensity scores for AIPW, or assuming bounded outcomes for TMLE) are used to mitigate this instability. By contrast, plug-in estimators never require truncation but lose efficiency. In response, we offer a unified approach that combines statistical efficiency with finite-sample stability, without additional heuristics.

Our framework reframes the problem of constructing a debiased ATE estimator as

a constrained optimization problem. Consider binary treatments (actions) A, covariates X, and potential outcomes $Y(1), Y(0)$ under treatment ($A = 1$) and control ($A = 0$), respectively. Under standard identification assumptions for the ATE, letting $(X, A, Y := Y(A)) \sim P$ for some distribution P, and also assuming $Y(0) := 0$ for simpler exposition, the ATE depends on the outcome model $\mu(X) := \mathbb{E}_P[Y \mid A = 1, X]$:

$$\text{ATE} = \mathbb{E}_P[Y(1) - Y(0)] = \mathbb{E}_P[Y(1)] = \mathbb{E}_P[\mu(X)].$$

Instead of fitting $\mu(\cdot)$ to minimize prediction error (as would be done for a standard plug-in estimator), we explicitly take into account the downstream causal estimation task by minimizing prediction error subject to the *constraint* that the first-order estimation error must be zero. For a given propensity score (treatment) model $\widehat{\pi}(X) \approx \mathbb{E}[A \mid X]$, we solve the following constrained learning problem over the class of outcome models $\widetilde{\mu}(\cdot)$ in the chosen model class $\mathcal{F}$:

$$\widehat{\mu}^C \in \underset{\widetilde{\mu} \in \mathcal{F}}{\operatorname{argmin}} \left\{ \text{PredictionError}(\widetilde{\mu}) : \text{First-order error of plug-in estimator from } \widetilde{\mu} \text{ is } 0 \right\}. \tag{1}$$

The resulting plug-in estimator $\frac{1}{n}\sum_{i=1}^n \widehat{\mu}^C(X_i)$ based on the constrained learning perspective (1) enjoys the usual fruits of first-order correction, such as semiparametric efficiency and double robustness, which we show in Section 6.

The constrained learning perspective (1) gives rise to a natural and performant estimation method, which we call the "C-Learner". Instead of adjusting an existing nuisance parameter estimate along a pre-specified direction to attain first-order correction (e.g., as in TMLE), we directly train the nuisance parameter to minimize prediction error subject to the constraint that the first-order estimation error must be zero. As we discuss in Section 5, we empirically observe that the C-Learner outperforms basic versions of one-step estimation and targeting without additional heuristics or approximations in challenging settings where there are covariate regions with little estimated overlap between treatment and control groups, where existing methods can exhibit instability. We observe C-Learner performs similarly to these versions of one-step estimation and targeting otherwise.

How does the C-Learner attain stable estimates, while basic versions of one-step estima-

tion and targeting are less stable in settings with low overlap? Inverse propensity weights (i.e. the reciprocal of the probability of treatment) can be extreme in settings with low overlap; basic versions of existing first-order correction methods are sensitive to extreme inverse propensity weights, while C-Learner is less sensitive to extreme inverse propensity weights as inverse propensity weights only appear in the constraint. The sensitivity of existing methods to extreme inverse propensity scores is visible through their definitions in Section 2.

Our constrained learning perspective also provides a way to unify versions of existing first-order correction methods like one-step estimation and targeting (Section 3.1). These versions of existing methods can be thought of as solving the optimization problem (1) with a restrictive model class given by augmenting a fitted nuisance estimator along a specific direction. For estimating the ATE, one version of one-step estimation uses an existing outcome model $\widehat{\mu}(X)$ plus an additive constant term $\mathcal{F} := \{\widehat{\mu}(\cdot)+\epsilon : \epsilon \in \mathbb{R}\}$, and one version of targeting uses an existing outcome model plus a canonical gradient-based covariate term $\mathcal{F} := \left\{\widehat{\mu}(\cdot) + \epsilon \frac{A}{\widehat{\pi}(\cdot)} : \epsilon \in \mathbb{R}\right\}$ where A is treatment and $\widehat{\pi}$ is a fitted propensity score (treatment) model. See Figure 1 for an illustration. In Section 6, we identify conditions under which a constrained learning estimator enjoys these results, and we verify that these one-step estimation and targeting methods also satisfy these necessary conditions.

Unlike traditional one-step estimators and targeted estimators, which apply a first-order correction to a single plug-in model, our C-Learner uses the entire chosen machine-learning model class $\mathcal{F}$ to compute that correction. In other words, instead of fixing a plug-in model and then adjusting it, we directly optimize over $\mathcal{F}$, which we expect to yield better finite-sample performance, which aligns with observations (Section 5). We implement the C-Learner using outcome models $\widehat{\mu}$ from linear models, and neural networks as described in Section 4 and evaluate on datasets chosen to match these model classes.

Contributions To summarize, we contribute the following:

1. The C-Learner (Section 3), a general method to estimate causal estimands that uses constrained optimization to attain asymptotic optimality (Section 6).

2. Practical instantiations of C-Learner for three different outcome model classes: linear

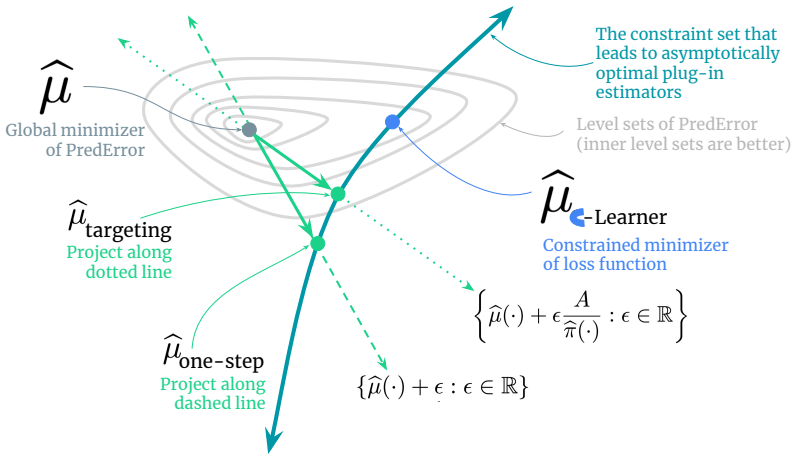

Figure 1. Schematic for how C-Learner is defined, compared to one-step estimation and targeting. $\widehat{\mu}$ is the unconstrained outcome model, which minimizes the prediction error on observed outcomes (gray ovals depict level sets for prediction error). $\widehat{\mu}_{\text{one-step}}$ and $\widehat{\mu}_{\text{targeting}}$ are different projections of $\widehat{\mu}$ onto the space of outcome models for which the corresponding plug-in estimators are asymptotically optimal. $\widehat{\mu}_{\text{C-Learner}}$ is the outcome model that minimizes prediction error on observed outcomes, subject to the plug-in estimator being asymptotically optimal (teal line).

models, gradient boosted trees, neural networks (Section 4).

3. Experiments using these three outcome model classes, on a range of settings, including with text covariates, that show C-Learner performs better than one-step estimation and targeting in settings with low overlap, and comparably otherwise (Section 5).

Related Works We defer a discussion of one-step estimation and targeting to Section 2. We defer a discussion of balancing estimators, which also achieve asymptotic optimality and have a connection to a "dual" version of C-Learner, to Section 3.3. Here, we discuss related works outside of one-step estimation, targeting, and balancing estimators. Machine learning approaches for nuisance estimation have received a lot of attention [1, 2, 3, 36, 25, 60]. These approaches can naturally be combined with first-order correction methods like one-step estimation and targeting, with recent work [42, 15] incorporating these ideas directly into model training through novel objectives and architectures. Like other first-order correction methods, C-Learner is orthogonal to and can also integrate innovations in nuisance estimation approaches; for example, in Section E.3.2, we demonstrate how C-Learner can effectively use Riesz representers learned by RieszNet [15].

2 Background

In the following, we illustrate the C-Learner in the context of the average treatment effect (ATE). We discuss extensions to other estimands in Section A.

2.1 Average Treatment Effect and Missing Outcomes

Consider binary treatments (actions) $A \in \{0,1\}$, covariates $X \in \mathbb{R}^d$, and potential outcomes $Y(1), Y(0) \in \mathcal{R}$ under treatment ($A=1$) and control ($A=0$), respectively. The key difficulty in causal inference is that we do not observe counterfactuals: we only observe the outcome $Y := Y(A)$ corresponding to the observed binary treatment. Let $Z := (X, A, Y)$. Based on i.i.d. observations $Z \sim P$, our goal is to estimate the average treatment effect $\psi(P) = P[Y(1) - Y(0)] := \int [Y(1) - Y(0)] dP$. Here P denotes both the joint probability measure and the expectation operator under this measure. We require standard identification conditions that make this goal feasible: (i) $Y = Y(A)$ (SUTVA), (ii) $(Y(1), Y(0)) \perp A \mid X$ (ignorability), and (iii) for some $\eta > 0$, $\eta \leq P(A = 1 \mid X) \leq 1 - \eta$ a.s. (overlap) [19].

To simplify our exposition, we assume $Y(0) := 0$ throughout so that we are estimating the mean of a censored outcome (censored when $A = 0$). This setting is also known as mean missing outcome [32], which is the focus of some of our experiments. In this case, we can write the ATE as a functional of the joint measure P, as

$$\psi(P) := P[Y(1)] = P\left[P[Y \mid A = 1, X]\right]. \tag{2}$$

The outcome model $\mu(X) := P[Y \mid A = 1, X]$ (shorthand for $\mu(A, X) := P[Y \mid A, X]$ with $\mu(0, X) := 0$) and propensity (treatment) model $\pi(X) := P(A = 1 \mid X)$ are key nuisance parameters; notably, they are high-dimensional in contrast to the single-dimensional ATE (2). Note that we can write $\psi(P) = P[\mu(1, X) - \mu(0, X)] = P[\mu(1, X)] = P[\mu(X)]$ in this setting. The corresponding nuisance estimators $\widehat{\mu}(X)$ and $\widehat{\pi}(X)$ can be implemented as ML models that are trained on held-out data.

2.2 First-Order Correction and Asymptotic Optimality

By analyzing the error from blindly trusting an ML-based estimate of the nuisance parameters to estimate the ATE (2), we can develop better estimation methods. We sketch intuition here and leave a more rigorous treatment to Section 6. We begin by noting that the joint distribution over the observed data can be decomposed as $P = P_{Y,A|X} \times P_X$. Since the marginal P_X can be simply estimated with an empirical distribution (plug-in), we focus on the error induced by approximating $P_{Y,A|X}$ using an estimate $\widehat{P}_{Y,A|X}$.

$$\begin{aligned} P_X[\widehat{\mu}(X)] - P_X[\mu(X)] &= P_X[\widehat{P}[Y \mid A=1, X]] - P_X[P[Y \mid A=1, X]] \\ &= \psi(P_X \times \widehat{P}_{Y,A|X}) - \psi(P_X \times P_{Y,A|X}). \end{aligned}$$

For the statistical functional $Q \mapsto \psi(Q)$, let φ be its canonical gradient (a.k.a. efficient influence function) with respect to $Q_{Y,A|X}$; without loss of generality, we require $Q[\varphi(Z;Q)|X] = 0$ for all Q. Then, a (distributional) first-order Taylor expansion gives

$$\begin{aligned} \psi(P_X \times \widehat{P}_{Y,A|X}) - \psi(P_X \times P_{Y,A|X}) &= \iint \varphi(Z;\widehat{P}) d(\widehat{P}_{Y,A|X} - P_{Y,A|X}) dP_X + R_2(\widehat{P}, P) \\ &= -\iint \varphi(Z;\widehat{P}) dP_{Y,A|X} dP_X + R_2(\widehat{P}, P) \\ &= -P[\varphi(Z;\widehat{P})] + R_2(\widehat{P}, P) \end{aligned} \tag{3}$$

where R_2 is a second-order remainder term. Conclude that the first-order error term of the plug-in approach $P_X[\widehat{\mu}(X)]$ is given by $-P[\varphi(Z;\widehat{P})]$.

A common approach in semiparametric statistics to better estimate $\psi(P)$ is to explicitly correct for this first-order error term. As we discuss later in Section 6, the first-order correction leads to asymptotic optimality properties like semiparametric efficiency—providing the shortest possible confidence interval—and double robustness—achieving estimator consistency even if only one of $\widehat{\mu}(X), \widehat{\pi}(X)$ is consistent.

For brevity of notation, for the rest of this section, we let φ denote a projected version of the canonical gradient, where empirically correcting for this projected version produces the same guarantees as empirically correcting for the full canonical gradient [57, 52].

First-Order Correction for the ATE For the ATE (2), it is well-known that

$$\varphi(Z;P) = \frac{A}{P[A=1|X]}(Y - P[Y|A=1,X]) = \frac{A}{\pi(X)}(Y-\mu(X)) \tag{4}$$

which in this work we do not derive. For an accessible primer on such derivations, see [32].

One-Step Estimation (Augmented Inverse Propensity Weighting, AIPW) [5] One of the most prominent first-order correction approaches modifies the plug-in estimator by moving the first-order error term to the left-hand side in the Taylor expansion (3). The resulting estimator achieves second-order error rates:

$$P\left[\widehat{\mu}(X) + \varphi(Z;\widehat{P})\right] - P[\mu(X)] = R_2(\widehat{P}, P).$$

Using the empirical distribution with samples $Z_1, \ldots, Z_N$ to approximate P in the one-step debiased estimator, we arrive at the augmented inverse propensity weighted estimator

$$\widehat{\psi}^{\text{AIPW}} := \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{\mu}(X_i) + \varphi(Z_i;\widehat{P})\right) = \frac{1}{n}\sum_{i=1}^{n}\widehat{\mu}(X_i) + \frac{1}{n}\sum_{i=1}^{n}\frac{A_i}{\widehat{\pi}(X_i)}(Y_i - \widehat{\mu}(X_i)). \tag{5}$$

Targeting (Targeted Maximum Likelihood Estimation) [57, 52] Targeting takes an alternative and more general approach to first-order correction. It commits to the use of the plug-in estimator, and constructs a tailored adjustment ("fluctuation") to the existing nuisance parameter estimate to set the first-order error to zero, where the magnitude of the adjustment is solved by maximizing likelihood (hence the name). The fluctuation takes place in the direction of a task-specific random variable ("clever covariate"), which takes different forms depending on the estimand. For illustrative purposes, in this discussion, we use focus on a formulation of targeting for unbounded outcomes under the squared loss function. In our experiments, we also include a commonly used variant of targeting that increases the stability of the estimator by assuming bounded outcomes, and modeling the outcomes with a logistic link. [1] We defer a deeper discussion of TMLE to Section F.

[1] The literature on targeting is large, with variants including regularization techniques and adaptive versions [57, 54, 52, 56]. We choose to focus on the formulation in Equation (6) for its simplicity. See the discussion on TMLE for bounded outcomes in Section 5.1, and Section F for additional discussion.

In the case of the ATE with $Y(0) := 0$ (so that $\widehat{\mu}(X) := \widehat{\mu}(1, X)$, as in Section 2.1) and general outcomes Y (including unbounded continuous Y), the form of the canonical gradient (4) motivates the use of $H(A, X) := \frac{A}{\widehat{\pi}(X)}$ as the clever covariate: targeting uses an adjusted nuisance estimate $\widehat{\mu}(A, X) + \epsilon^\star H(A, X)$ in place of $\widehat{\mu}(A, X)$ in the plug-in for $\psi(P) = P[\mu(1, X)]$, where $\epsilon^\star$ is chosen to solve the targeted maximum likelihood problem

$$\epsilon^\star := \underset{\epsilon \in \mathbb{R}}{\operatorname{argmin}} \quad \frac{1}{n} \sum_{i=1}^{n} A_i \left(Y_i - \widehat{\mu}(A_i, X_i) - \epsilon \frac{A_i}{\widehat{\pi}(X_i)} \right)^2. \tag{6}$$

From the KKT conditions, the solution removes the finite-sample estimate of the first-order error term (3) of the plug-in estimator using $\widehat{\mu}(A, X) + \epsilon^\star \frac{A}{\widehat{\pi}(X)}$. Solving for $\epsilon^\star$, we obtain

$$\epsilon^\star = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{A_i}{\widehat{\pi}(X_i)^2} \right)^{-1} \frac{1}{n} \sum_{i=1}^{n} \frac{A_i}{\widehat{\pi}(X_i)} (Y_i - \widehat{\mu}(X_i)). \tag{7}$$

Thus, we arrive at an explicit formula for the targeted maximum likelihood estimator

$$\begin{aligned} \widehat{\psi}^{\mathrm{TMLE}} &:= \frac{1}{n} \sum_{i=1}^{n} \left(\widehat{\mu}(1, X_i) + \epsilon^\star H(1, X_i) \right) \frac{1}{n} \sum_{i=1}^{n} \left(\widehat{\mu}(X_i) + \epsilon^\star \frac{1}{\widehat{\pi}(X_i)} \right) \\ &= \frac{1}{n} \sum_{i=1}^{n} \widehat{\mu}(X_i) + \frac{\sum_{i=1}^{n} \frac{1}{\widehat{\pi}(X_i)}}{\sum_{i=1}^{n} \frac{A_i}{\widehat{\pi}(X_i)^2}} \cdot \frac{1}{n} \sum_{i=1}^{n} \frac{A_i}{\widehat{\pi}(X_i)} (Y_i - \widehat{\mu}(X_i)). \end{aligned} \tag{8}$$

Targeted Regularization [15, 42] Instead of modifying the outcome model in the direction of the clever covariate in a post-processing step (6), we can instead regularize the usual training objective for the outcome model using a similar adjustment term throughout training, an approach referred to as *targeted regularization*, where it is demonstrated on e.g. neural networks learned via stochastic gradient-based optimization.[2]

[2] An additional difference between TMLE and targeted regularization is the data splits involved. In TMLE, the targeting objective (6) for calculating ϵ^* uses the same data that is used for plug-in estimation (the first line in (8)), while in targeted regularization, the targeting objective is applied to the dataset used for training $\widehat{\mu}$, which has not explicitly described above, as so far, $\widehat{\mu}, \widehat{\pi}$ are taken as given. These data splits are often different; see Section 4.1 for more discussion on data splits.

3 Constrained Learning Framework

The aforementioned approaches to first-order correction take the fitted nuisance estimate as given, and make adjustments to either the estimator (one-step estimation (5)) or the nuisance estimate (targeting (6)). In this section, we propose the *constrained learning framework* to first-order correction where we train the nuisance parameter to be the best nuisance estimator subject to the *constraint* that the first-order error term (3) is zero.

Our method, which we call the C-Learner, is a general method for adapting machine learning models to explicitly consider the semiparametric nature of the downstream task during training. For one-step estimation and targeting, in the sections before, we would use a fitted outcome model $\widehat{\mu}$ that minimizes the squared prediction loss (or equivalently maximizes likelihood, assuming outcomes are Gaussian), with

$$\widehat{\mu} \in \operatorname*{argmin}_{\widetilde{\mu} \in \mathcal{F}} P_{\text{train}}[A(Y - \widetilde{\mu}(X))^2], \tag{9}$$

where the loss minimization is performed over an auxiliary training data split P_{train} which may be separate from the main sample $(X_i, A_i, Y_i)_{i=1}^n$ on which estimators are calculated. Instead, the C-Learner solves the constrained optimization problem

$$\widehat{\mu}^C \in \operatorname*{argmin}_{\widetilde{\mu} \in \mathcal{F}} \left\{ P_{\text{train}}[A(Y - \widetilde{\mu}(X))^2] : \frac{1}{n} \sum_{i=1}^n \frac{A_i}{\widehat{\pi}(X_i)} (Y_i - \widetilde{\mu}(X_i)) = 0 \right\}, \tag{10}$$

$$\widehat{\psi}^{\text{C-Learner}} := \frac{1}{n} \sum_{i=1}^n \widehat{\mu}^C(X_i) \tag{11}$$

so that the constraint in Equation (10) is applied to the same data used in the plug-in (11). Observe that the first-order error term being 0 is a sort of balancing constraint: it forces the inverse propensity weighted plug-in $\frac{1}{n}\sum_{i=1}^n A_i \widehat{\mu}^C(X_i)/\widehat{\pi}(X_i)$ to equal the inverse propensity weighted estimator $\frac{1}{n}\sum_{i=1}^n A_i Y_i/\widehat{\pi}(X_i)$ [19]. This constrained optimization is done over model class $\mathcal{F}$, which can be chosen to be suitable for the problem setting.

The constraint (10) ensures that the corresponding plug-in estimator (11) has a finite-sample estimate of the first-order error term (3) of zero. The training objective for $\widehat{\mu}^C$ thus optimizes for the best outcome model fit using the training data, subject to the constraint that the plug-in estimator is asymptotically optimal. In practice, there are several com-

putational approaches to (approximately) solve the stochastic optimization problem (10) depending on the function class $\mathcal{F}$. In Section 4, we describe how to instantiate C-Learner using linear models, gradient boosted regression trees, and neural networks. We empirically demonstrate these in Section 5.

Like other first-order correction methods, the C-Learner can be implemented with cross-fitting [13], in which models (nuisance parameters) are evaluated on data splits that are separate from the data splits on which models are trained; we discuss data splitting further in Section 4.1. By virtue of being debiased, the C-Learner enjoys the same asymptotic properties as the AIPW and TMLE. See Section 6 for a formal treatment.

Theorem 1 (Informal)**.** *The cross-fitted version of* $\widehat{\psi}^{\text{C-Learner}}$ *is semiparametrically efficient and doubly robust.*

Although we illustrate the C-Learner in the ATE setting for simplicity, our approach generalizes to other estimands that are continuous linear functionals of outcome μ (Section A).

3.1 Relationship With Other Approaches to First-Order Debiasing

The constrained learning framework provides a unifying perspective to existing approaches to first-order correction. First, a variant of AIPW (5) can also be thought of as a C-Learner over a very restricted class $\mathcal{F}$ of outcome models. For a given trained outcome model $\widehat{\mu}$, consider the constrained optimization problem (10) over the model class $\mathcal{F} := \{\widehat{\mu}(X) + \epsilon : \epsilon \in \mathbb{R}\}$. In order to satisfy the constraint (10), we must have $\epsilon^* = \left(\frac{1}{n}\sum_{i=1}^n \frac{A_i}{\widehat{\pi}(X_i)}\right)^{-1} \frac{1}{n}\sum_{i=1}^n \frac{A_i}{\widehat{\pi}(X_i)}(Y_i - \widehat{\mu}(X_i))$, which gives this special case of the C-Learner,

$$\begin{aligned}\widehat{\psi}^{\text{AIPW-SN}} &= \frac{1}{n}\sum_{i=1}^n \left(\widehat{\mu}(X_i) + \epsilon^\star\right) \\ &= \frac{1}{n}\sum_{i=1}^n \widehat{\mu}(X_i) + \left(\frac{1}{n}\sum_{i=1}^n \frac{A_i}{\widehat{\pi}(X_i)}\right)^{-1} \frac{1}{n}\sum_{i=1}^n \frac{A_i}{\widehat{\pi}(X_i)}(Y_i - \widehat{\mu}(X_i)). \end{aligned} \tag{12}$$

We refer to this as a *self-normalized AIPW*. This is the same as the AIPW (5) aside from a normalization term $\frac{1}{n}\sum_{i=1}^n \frac{A_i}{\widehat{\pi}(X_i)}$ that has expectation 1 if we use the true propensity score π instead of $\widehat{\pi}$. We observe empirically (Section 5) that the self-normalized AIPW (12),

which can be thought of as a variant of AIPW motivated by our constrained optimization perspective, enjoys better finite-sample performance than the standard AIPW (5).

For general outcome variables (including unbounded continuous outcome variables), we show that a basic version of targeting (8) can be viewed as a C-Learner over a specific function class. By inspection, the first-order condition in (6) is given by

$$\frac{1}{n}\sum_{i=1}^{n}\frac{A_i}{\widehat{\pi}(X_i)}\left(Y_i-\widehat{\mu}(X_i)-\epsilon\frac{A_i}{\widehat{\pi}(X_i)}\right)=0. \tag{13}$$

Thus, this version of targeting is a C-Learner where a pre-trained outcome model $\widehat{\mu}(A,X)$ is perturbed along a specific direction to become $\widehat{\mu}^C(A,X) := \widehat{\mu}(A,X)+\epsilon\frac{A}{\widehat{\pi}(X)}$.[3] Reframing the constrained optimization problem (10) with model class $\mathcal{F} := \left\{\widehat{\mu}(X)+\epsilon\frac{A}{\widehat{\pi}(X)} : \epsilon\in\mathbb{R}\right\}$ thus provides a new way to view this basic version of targeting (8).

For clarity, we do not use "C-Learner" to refer to self-normalized AIPW (12) or to targeting (6), even though they can be thought of as C-Learners over a restricted model class.

3.2 C-Learner is Numerically Stable, Without Additional Heuristics

The aforementioned approaches to first-order correction rely on adding estimated ratios ($A/\widehat{\pi}(X)$ for one-step estimation (5)) or fluctuating the outcome model $\widehat{\mu}$ in a specific direction (along $A/\widehat{\pi}(X)$ for targeting (6)) in order to de-bias plug-in estimators (Section 2.2). In contrast, C-Learner achieves asymptotic optimality without using limited model classes $\mathcal{F}$ as in one-step estimation and targeting (Section 3.1): it simply trains the nuisance parameter $\widehat{\mu}^C(\cdot)$ so the plug-in estimator $\frac{1}{n}\sum_{i=1}^{n}\widehat{\mu}^C(X_i)$ satisfies the criterion for asymptotic optimality in the most direct way possible, while using the entirety of the chosen model class.

In settings with regions of low overlap between treatment and control, the probability of treatment $\widehat{\pi}(X)$ can be very close to 0 for some values of X, so that $1/\widehat{\pi}(X)$ can be extremely large, causing numerical instability in one-step estimation and targeting. While there are variations for both AIPW and TMLE that are more stable, e.g., by self-normalizing propensity weights, truncating $\widehat{\pi}$ to avoid extreme values in AIPW, or assuming Y is bounded and modeling Y as a (scaled) logistic function of (A,X) in TMLE (see Sec-

[3] Recall that for simplicity, we assume $Y(0) := 0$.

tion 5.1), C-Learner can avoid instability simply by having this $1/\widehat{\pi}(X)$ appear only in the constraint in the constrained optimization, rather than an additive term to the estimator.

Additionally, simple plug-in estimators of outcome models from their chosen model classes have been observed to be numerically stable (e.g., [31]) and empirically, the C-Learner appears to inherit these benefits (Section 5).

3.3 A "Dual" C-Learner and Connections to Covariate Balancing

The formulation of Equation (10) starts by fitting and fixing a propensity score $\widehat{\pi}$, and then fitting a constrained outcome model $\widehat{\mu}^C$ to maximize likelihood, subject to a constraint that depends on $\widehat{\pi}$ to ensure efficiency of the direct plug-in estimator that uses $\widehat{\mu}^C$. One could alternatively first fit an outcome model $\widehat{\mu}$, and then fit a propensity model $\widehat{\pi}^C$ to maximize likelihood, also subject to a constraint to ensure efficiency of the resulting estimator, which we call the "dual" C-Learner:

$$\widehat{\pi}^C \in \underset{\tilde{\pi}\in\mathcal{F}_\pi}{\operatorname{argmin}} \left\{ P_{\text{train}}[\tilde{\pi}(X)^A(1-\tilde{\pi}(X))^{1-A}] : \frac{1}{n}\sum_{i=1}^{n}\left(1-\frac{A_i}{\tilde{\pi}(X_i)}\right)\widehat{\mu}(X_i) = 0 \right\},$$

$$\widehat{\psi}^{\text{C-Learner-Dual}} := \frac{1}{n}\sum_{i=1}^{n}\frac{A_i}{\widehat{\pi}^C(X_i)}Y_i. \tag{14}$$

The constraint is chosen analogously to the constraint for the usual C-Learner, and we show how such a constraint leads to semiparametric efficiency in Section D.1.

The rest of this paper focuses on the C-Learner in Equation (10), and we include the dual C-Learner primarily to relate to existing literature. This "dual" C-Learner formulation is connected to covariate balancing propensity scores (CBPS) [28]. Specifically, the covariate balancing condition from [28] for estimating the ATE can be written as

$$\mathbb{E}\left[\frac{Af(X)}{\pi(X)} - \frac{(1-A)f(X)}{1-\pi(X)}\right] = 0 \tag{15}$$

where $f(X)$ can be any function of X. For example, one practical choice could be $f(X) = X$ so that ensuring the condition in (15) results in balancing each covariate dimension.

As our paper focuses on the mean missing outcome setting (equivalent to the ATE with

$Y(0) = 0$ for cleaner notation), we focus on a corresponding balancing condition, i.e. [44]

$$\mathbb{E}\left[\left(1 - \frac{A}{\pi(X)}\right) f(X)\right] = 0, \tag{16}$$

again for any choice of f; note that if (16) holds for both A and $1 - A$, i.e. that

$$\mathbb{E}[f(X)] = \mathbb{E}\left[\frac{A}{\pi(X)} f(X)\right] \quad \text{and} \quad \mathbb{E}[f(X)] = \mathbb{E}\left[\frac{1-A}{1-\pi(X)} f(X)\right] \tag{17}$$

then (15) must also hold. A finite-sample counterpart of (16) can also be written [44] as

$$\frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{A_i}{\widehat{\pi}(X_i)}\right) f(X) = 0. \tag{18}$$

Notably, this condition is exactly the constraint in Equation (14) when we take f to be $\widehat{\mu}$. CBPS can then be used to enforce this constraint if, for example, μ is linear in some basis of functions $\{f_1, \ldots, f_J\}$, and balance is enforced for all elements in this basis. In this way, covariate balancing can result in semiparametric efficiency.

Related works Tan [44] produces semiparametrically efficient IPW-based estimators, similar to a "dual" C-Learner, but their methods are analogous to targeting, as they learn an "extended" propensity score that is a parametric fluctuation from an initial propensity score model, where the magnitude of the fluctuation is determined by maximizing likelihood. Unlike targeting, their estimators have additional constraints to ensure boundedness of their estimator, improved local efficiency (better variance for a given fixed estimated outcome model; this is outside of the scope of our work), and double robustness. Their theoretical results focus on parametric nuisance models and notions of efficiency when certain nuisance parameters are well-specified vs. not, unlike in our setting, but their estimators also achieve the semiparametric efficiency bound.

Zhao [61] proposes a loss function to minimize for learning a nonparametric propensity model, where the first-order conditions of the loss are the balancing conditions in CBPS. They also show their corresponding inverse propensity weighted estimators are efficient, but under different assumptions: Zhao [61] shows efficiency using a sieve approach with a growing basis of functions to model the propensity score through a logistic link, while we

assume standard non-parametric rates of convergence for nuisance parameters.

4 Methodology

The constrained learning framework can be instantiated in many ways depending on the function class $\mathcal{F}$ for outcome models. Concretely, we illustrate the versatility of the C-Learner by presenting approximate solution methods to the constrained optimization problem (10) for linear models, gradient boosted regression trees, and deep neural networks. We empirically demonstrate these instantiations in Section 5.[4]

4.1 Data Splitting

We recommend sample splitting, where we split the data so that nuisance estimators (e.g. $\widehat{\pi}, \widehat{\mu}, \widehat{\mu}^C$) are fitted on a training (auxiliary) fold and evaluated to form a causal estimator on an evaluation (main) fold. Let P_{train} be the split on which nuisance parameters such as the outcome model or the propensity score $\widehat{\pi}(\cdot)$ are trained, and let P_{val} be the split on which we perform model selection on them. We use a separate split P_{eval} to evaluate our final causal estimators. Our constrained learning framework (10) can thus be rewritten as

$$\begin{aligned} &\widehat{\psi}^{\text{C-Learner}} := P_{\text{eval}}\left[\widehat{\mu}^C(X)\right] \qquad \text{where} \\ &\widehat{\mu}^C \in \underset{\widetilde{\mu}\in\mathcal{F}}{\operatorname{argmin}} \left\{ P_{\text{train}}[A(Y-\widetilde{\mu}(X))^2] : P_{\text{eval}}\left[\frac{A}{\widehat{\pi}(X)}(Y-\widetilde{\mu}(X))\right] = 0 \right\}. \end{aligned} \tag{19}$$

For simplicity, we let $P_{\text{eval}} = P_{\text{val}}$ except where otherwise specified. *Cross-fitting* with K folds [13] refers to the following sample splitting setup designed to utilize all of the data: first, we split the data into K folds. Then, to evaluate on the kth fold in $P_{\text{eval}}[\widehat{\mu}^C(X)]$, we train nuisance parameters on all but the kth fold in the data. We repeat this for all K folds and average the results, so that the final estimator utilizes model evaluations over the entire dataset. This setup for C-Learner is treated more formally in Section 6.

[4] Other frameworks [34] exist for constructing constrained outcome models, but do not restrict outcome models to $\mathcal{F}$; we exclude these from our analysis.

4.2 Linear Models

When outcome models are linear functions of X, the constrained learning problem (19) has an analytic solution. Using $\vec{\cdot}$ to denote stacked observations, define

$$\begin{bmatrix} \vec{Y}_{\text{train}} \\ \vec{X}_{\text{train}} \\ \vec{H}_{\text{train}} \end{bmatrix} := \begin{bmatrix} \{Y_i\}_{i\in\mathcal{I}_{\text{train}}} \\ \{X_i\}_{i\in\mathcal{I}_{\text{train}}} \\ \left\{\frac{A_i}{\widehat{\pi}(X_i)}\right\}_{i\in\mathcal{I}_{\text{train}}} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \vec{Y}_{\text{eval}} \\ \vec{X}_{\text{eval}} \\ \vec{H}_{\text{eval}} \end{bmatrix} := \begin{bmatrix} \{Y_i\}_{i\in\mathcal{I}_{\text{eval}}} \\ \{X_i\}_{i\in\mathcal{I}_{\text{eval}}} \\ \left\{\frac{A_i}{\widehat{\pi}(X_i)}\right\}_{i\in\mathcal{I}_{\text{eval}}} \end{bmatrix}$$

where $\mathcal{I}_{\text{train}} = \{i \in \text{train} : A_i = 1\}$ and $\mathcal{I}_{\text{eval}} = \{i \in \text{eval} : A_i = 1\}$ are indices with observations in each data split. The constrained learning problem (19) can be rewritten as

$$\widehat{\theta}^C = \underset{\theta}{\text{argmin}} \left\{ \frac{1}{2}\|\vec{Y}_{\text{train}} - \vec{X}_{\text{train}}\theta\|^2 : \ \vec{H}_{\text{eval}}^\top(\vec{Y}_{\text{eval}} - \vec{X}_{\text{eval}}\theta) \right\}.$$

The KKT conditions characterize the primal-dual optimum $(\widehat{\theta}^C, \widehat{\lambda})$

$$\widehat{\theta}^C = (\vec{X}_{\text{train}}^\top \vec{X}_{\text{train}})^{-1}\vec{X}_{\text{train}}^\top(\vec{Y}_{\text{train}} + \widehat{\lambda}\vec{H}_{\text{train}}) \text{ where } \widehat{\lambda} = \frac{\vec{H}_{\text{eval}}^\top(\vec{Y}_{\text{eval}} - \vec{Y}_{\text{ols}})}{\vec{H}_{\text{eval}}^\top \vec{X}_{\text{eval}}(\vec{X}_{\text{train}}^\top \vec{X}_{\text{train}})^{-1}\vec{X}_{\text{train}}^\top \vec{H}_{\text{train}}}$$

and $\vec{Y}_{\text{ols}} := \vec{X}_{\text{train}}^\top(\vec{X}_{\text{train}}^\top \vec{X}_{\text{train}})^{-1}\vec{X}_{\text{train}}^\top \vec{Y}_{\text{train}}$. Note that $\widehat{\theta}^C$ is the OLS with respect to the pseudo-label $\vec{Y}_{\text{train}} + \widehat{\lambda}\vec{H}_{\text{train}}$ and the dual variable shifts the observed outcomes in the direction of $\vec{H}_{\text{train}}$ similar to targeting (6), but with additional reweighting using covariates. When using our framework restricted to linear function classes, we obtain a new estimator that, to the best of our knowledge, cannot be recovered by existing methods. In Section 5.1, we demonstrate that this new estimator improves upon existing methods.

4.3 Gradient Boosted Regression Trees

We consider outcome models that are gradient boosted regression trees. The gradient boosting framework [22] iteratively estimates the functional gradient g_j of the loss function evaluated on the current function estimate $\widehat{\mu}_j$. For the standard MSE loss $\ell(\mu; X, A, Y) := A(Y - \mu(X))^2$, we use weak learners in $\mathcal{G}$ (e.g., shallow decision trees) to compute

$$\widehat{g}_{j+1} \in \underset{g\in\mathcal{G}}{\text{argmin}}\, P_{\text{train}}[(A(g_j(X,Y) - g(X))^2] \quad \text{where} \quad g_j(X, Y; \widehat{\mu}_j) := \frac{\partial}{\partial\mu}\ell(\mu; X, Y)\Big|_{\mu=\widehat{\mu}_j}.$$

Algorithm 1 C-Learner with Gradient Boosted Regression Trees

Input: learning rate η, max trees J and K, $\widehat{\mu}_0 := 0$
for $j = 0, 2, \ldots, J-1$ **do**
    Modify functional gradient $g_j = Y - \widehat{\mu}_j(X)$ to $\widetilde{g}_j = g_j + \epsilon_j^\star \cdot A/\widehat{\pi}(X)$ where $\epsilon_j^\star := \frac{P_{\text{train}}[(Y-\widehat{\mu}_j(X))\cdot A/\widehat{\pi}(X)]}{P_{\text{train}}[A/\widehat{\pi}(X)^2]}$ as in (20)
    Compute $\widehat{g}_j = \text{argmin}_{g\in\mathcal{G}} P_{\text{train}}[A(\widetilde{g}_j - g(X))^2]$ and update $\widehat{\mu}_{j+1} = \widehat{\mu}_j - \eta\widehat{g}_j$
end for
for $k = 0, 2, \ldots, K-1$ **do**
    Compute gradient $\widetilde{g}_k = \epsilon_{J+k}^\star \cdot A/\widehat{\pi}(X)$ where $\epsilon_{J+k}^\star = \frac{P_{\text{eval}}[(Y-\widehat{\mu}_{J+k}(X))\cdot A/\widehat{\pi}(X)]}{P_{\text{eval}}[A/\widehat{\pi}(X)^2]}$
    Compute $\widehat{g}_k = \text{argmin}_{g\in\mathcal{G}} P_{\text{eval}}[A(\widetilde{g}_k - g(X))^2]$ and update $\widehat{\mu}_{J+k+1} = \widehat{\mu}_{J+k} - \eta\widehat{g}_k$
end for
Return final outcome model $\widehat{\mu}^C_{\text{XGB}} := \widehat{\mu}_{J+K}$

and set $\widehat{\mu}_{j+1} = \widehat{\mu}_j - \eta\widehat{g}_{j+1}$ for some step size η. As we assume $Y := 0$ in our setting, we let $\mu(X)$, $g_j(X,Y)$, and $g(X,Y)$ be shorthand for $\mu(A,X)$, $g_j(X,A,Y)$, and $g(X,A,Y)$ when setting $A = 1$, and we let $\mu(0,X), g_j(0,X), g(0,X) = 0$. The process repeats J times until a maximum number of weak learners are fitted or an early stopping criterion is met.

Constrained Gradient Boosting We want to minimize the squared loss subject to the constraint (10) and propose a two-stage procedure. First, we perform gradient boosting where instead of the functional gradient of the loss g_j, we use a modified version

$$\widetilde{g}_j := g_j + \epsilon_j^\star \cdot \frac{A}{\widehat{\pi}(X)} \quad \text{where} \quad \epsilon_j^\star = \underset{\epsilon}{\text{argmin}} \left\{ P_{\text{train}} \left[A \left(Y - \widehat{\mu}_j(X) - \epsilon \cdot \frac{A}{\widehat{\pi}(X)} \right)^2 \right] \right\} \tag{20}$$

given by the targeting objective (6) applied to $\widehat{\mu}_j$ on the dataset P_{train}. The modification $\widetilde{g}_j$ allows subsequent weak learners to be fit in a direction that reduces the loss *and* makes the plug-in estimator closer to satisfying our constraint on P_{train}. To ensure the constraint (19) is satisfied on P_{eval}, the second stage fits weak learners to the gradient of constraint violation below:

$$\widetilde{g}_k := \epsilon_{J+k}^\star \cdot \frac{A}{\widehat{\pi}(X)} \quad \text{where} \quad \epsilon_{J+k}^\star = \underset{\epsilon}{\text{argmin}} \left\{ P_{\text{eval}} \left[A \left(Y - \widehat{\mu}_{J+k}(X) - \epsilon \cdot \frac{A}{\widehat{\pi}(X)} \right)^2 \right] \right\}.$$

We summarize the method above in pseudo-code in Algorithm 1, which we implement using the XGBoost package with custom objectives [12]. Hyperparameters for the first stage (learning rate η, and other properties of the weak learners such as max tree depth) are selected to have the lowest MSE loss on P_{val}. Other hyperparameters such as max number of trees J and K, may be set on P_{val}, or alternatively, by early stopping, with evaluation on different splits within P_{train}. These hyperparameters are re-used in the second stage.

4.4 Neural Networks

When outcome models are neural networks $\widehat{\mu}_\theta(x)$ with weights θ, we consider the usual MSE loss with a Lagrangian regularizer for the constraint (19)

$$R(\theta) = P_{\text{train}}\left[A(Y - \widehat{\mu}_\theta(X))^2\right] + \lambda \cdot P_{\text{eval}}\left[\frac{A}{\widehat{\pi}(X)}(Y - \widehat{\mu}_\theta(X))\right]^2.$$

We optimize the objective using stochastic gradient methods, where we take mini-batches of the training set to approximate the gradient of the first term and take a full-batch gradient on the evaluation set for the second term. At the end of every training epoch, the constraint on P_{eval} is enforced *exactly* by adjusting the constant bias term θ_{bias} in the neural network:

$$\theta_{\text{bias}} \leftarrow \theta_{\text{bias}} + \left(P_{\text{eval}}\left[\frac{A}{\widehat{\pi}(X)}\right]\right)^{-1} P_{\text{eval}}\left[\frac{A}{\widehat{\pi}(X)}(Y - \widehat{\mu}_\theta(X))\right]. \tag{21}$$

We choose to stop training at the epoch that minimizes $P_{\text{val}}\left[A(Y - \widehat{\mu}_\theta(X))^2\right]$, the MSE loss on the validation split. We consider two options for choosing hyperparameters (e.g. learning rate, λ). The first option is to minimize MSE on P_{val}. Second, since a small bias shift indicates a regularizer that is successful, we choose among hyperparameters with reasonable MSE loss on P_{val} the one that minimizes the size of the bias shift (21) in the first epoch. While the first method is more standard, the second encourages satisfying the constraint over the course of training, to avoid big jumps in optimization. Model selection for nuisance parameters is an active area of research [25, 39, 41]; we explore these in Section 5.2.

5 Experiments

In this section, we present a series of experiments to demonstrate how C-Learner[5] is flexible across data types and model classes. We also show that C-Learner achieves good empirical performance among all these settings, especially variants with low overlap. First, in Section 5.1, we consider well-studied tabular simulated settings where existing debiasing methods (e.g. AIPW) are known to perform poorly [31, 38]. Surprisingly, C-Learner is the only debiased method that performs comparably with the naive plug-in estimator, without additional heuristics or assumptions. We show that the same holds for a more flexible function class using gradient-boosted trees for the outcome models. We also show that the dual C-Learner (Section 3.3) performs at least as well as covariate balancing methods.

To test the scalability of our approach, in Section 5.2 we construct and study a high-dimensional setting with text features, where we fine-tune a language model [40] under our constrained learning framework. Again, we observe that C-Learner outperforms one-step estimation and targeting, especially in settings with low overlap. We summarize the results of our experiments in the above two settings in Figure 2.

Lastly, in Section E.3, we study a common tabular dataset (Infant Health and Development Program [10]), where the C-Learner is implemented with gradient boosted regression trees matches the performance of one-step estimation and targeting.

5.1 Tabular Dataset from Kang and Schafer [31]

We start with the synthetic tabular setting constructed by Kang and Schafer [31], who demonstrate empirically that the direct method (a naive plug-in estimator with a linear outcome model, labeled as "OLS" in [31]) achieves better performance than asymptotically optimal methods in their setting. Robins et al. [38] note that settings in which some subpopulations have a much higher probability of being treated than others, like in Kang and Schafer [31], are very challenging for existing asymptotically optimal methods. Thus, we evaluate the C-Learner on this well-known and challenging setting.

Our goal is to estimate $\psi(P) := P[Y(1)] = P[P[Y|A = 1]] = P[\mu(X)]$ from data

[5]Here we use "C-Learner" to refer to estimators using solutions to the constrained optimization problem in Section 3 for a chosen model class, even though other methods can also be thought of as C-Learners, as discussed in Section 3.1.

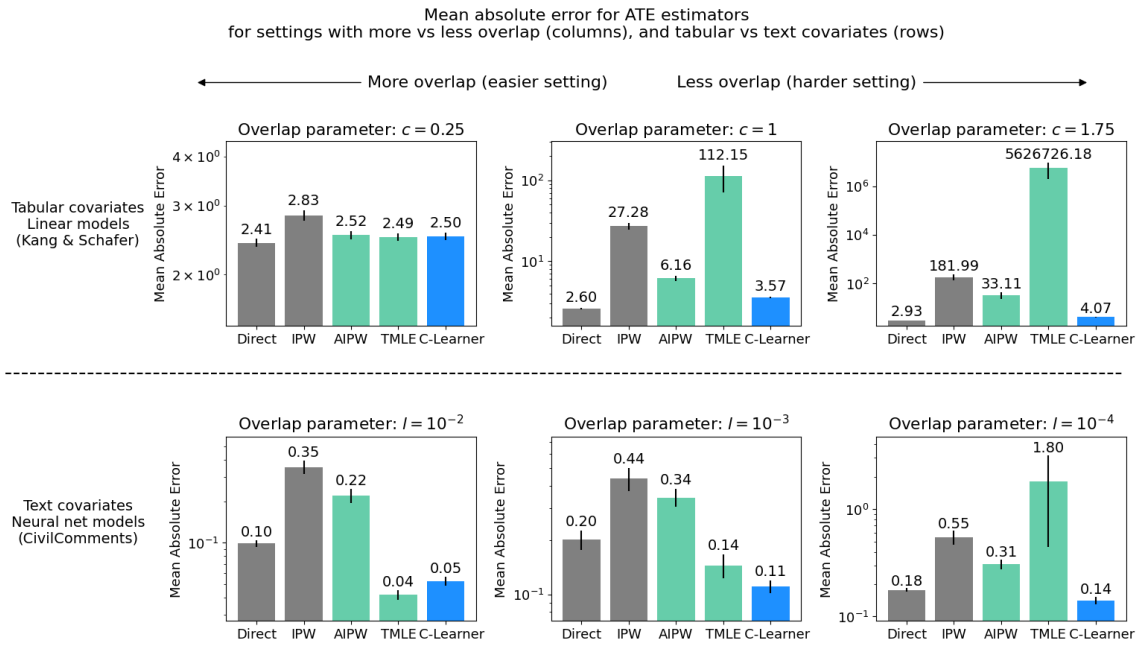

Figure 2. Comparison of mean absolute error of various estimators, in two types of settings (tabular covariates in the top row with sample sizes of $N = 200$ from Section 5.1, and text covariates in the bottom row with sample sizes of $N = 2000$ from Section 5.2), and with settings with a range of difficulty (more overlap in the left columns, less overlap in the right columns). Error bars represent ± 1 standard error, over 1000 and 100 dataset draws for tabular and text settings, respectively. C-Learner performs well, even in settings with low overlap. "Direct" refers to the naive plug-in estimator of the best fit outcome model $\frac{1}{n}\sum_{i=1}^{n}\widehat{\mu}(X_i)$, "IPW" refers to inverse propensity score weighting, "AIPW" refers to the augmented IPW, "TMLE" refers to targeted maximum likelihood estimation. AIPW, TMLE, and C-Learner are asymptotically optimal.

(X, A, AY) (assuming $Y(0) := 0$). The *true* outcome and treatment mechanisms depend on covariates $\xi \sim N(0, I) \in \mathbb{R}^4$ and $\varepsilon \sim N(0, 1)$ via $Y = 210 + 27.4\xi_1 + 13.7\xi_2 + 13.7\xi_3 + 13.7\xi_4 + \varepsilon$ and $\pi(\xi) = \frac{\exp(-\xi_1 + 0.5\xi_2 - 0.25\xi_3 - 0.1\xi_4)}{1+\exp(-\xi_1 + 0.5\xi_2 - 0.25\xi_3 - 0.1\xi_4)}$. Here, $P[Y(1)] = P[Y \mid A = 1] = 200$ and $P[Y] = 210$ so that a naive average of the treated units is biased by -10. For a random sample of 100 data points, the true propensity score can be as low as 1% and as high as 95%. We focus on the misspecified setting, where instead of observing ξ, the modeler observes $X_1 = \exp(\xi_1/2),\ X_2 = \xi_2/(1 + \exp(\xi_1)) + 10,\ X_3 = (\xi_1\xi_3/25 + 0.6)^3,\ X_4 = (\xi_2 + \xi_4 + 20)^2$. Next, we demonstrate the instantiations of the C-Learner with two model classes: linear models and gradient boosted regression trees.

	(a) $N = 200$				(b) $N = 1000$			
Method	Bias		Mean Abs Err		Bias		Mean Abs Err	
Direct	-0.00	(0.10)	2.60	(0.06)	-0.43	(0.04)	1.17	(0.03)
IPW	22.10	(2.58)	27.28	(2.53)	105.46	(59.84)	105.67	(59.84)
IPW-SN	3.36	(0.29)	5.42	(0.26)	6.83	(0.33)	7.02	(0.32)
AIPW	-5.08	(0.474)	6.16	(0.46)	-41.37	(24.82)	41.39	(24.82)
AIPW-SN	-3.65	(0.20)	4.73	(0.18)	-8.35	(0.43)	8.37	(0.43)
TMLE	-111.59	(41.07)	112.15	(41.07)	-17.51	(3.49)	17.51	(3.49)
C-Learner	**-2.45**	(0.12)	**3.57**	(0.09)	**-4.40**	(0.07)	**4.42**	(0.07)
TMLE-L	-2.06	(0.10)	3.10	(0.07)	-3.68	(0.06)	3.68	(0.05)
C-Learner-L	***-0.792***	(0.11)	***2.89***	(0.07)	***-1.89***	(0.07)	***2.27***	(0.06)

Table 1. Comparison of estimators on misspecified Kang and Schafer [31] settings in 1000 tabular simulations, for linear outcome models (Section 5.1.1). Asymptotically optimal methods are listed beneath the solid horizontal divider. Asymptotically optimal methods that use a logistic link are below the dashed horizontal divider. We highlight the best-performing *asymptotically optimal* method, not including the logistic link, in **bold**; we highlight the best-performing *asymptotically optimal* method *overall* in ***bold-italic***. Standard errors are displayed in parentheses to the right of the point estimate.

5.1.1 Linear Outcome Models

Linear outcome models are appealing due to their simplicity and interpretability, and are a fundamental tool for both theorists and practitioners. Here, we fit a linear model $\widehat{\mu}(X)$ on covariates X to predict the potential outcome Y. The outcome models we employ achieve an R^2 value close to 0.99, indicating a high degree of fit to the observed data. We also fit logistic propensity score models $\widehat{\pi}$; the ROC is approximately 0.75, suggesting reasonable classification performance. Following the original study by Kang and Schafer [31], we do not distinguish between data splits; there is only one split ($P_{\text{train}} = P_{\text{val}} = P_{\text{eval}}$ in Section 4.1).

We present a comparison of ATE estimators that use $\widehat{\mu}, \widehat{\pi}$ as described above in Table 1. In this table, "Direct" refers to the plug-in with the outcome model trained as usual ("OLS" in [31]), "IPW-SN" [31] refers to self-normalized IPW (a.k.a. Hajek estimator [19]), and "AIPW-SN" refers to self-normalized AIPW (see Section 3.1). C-Learner performs best out of asymptotically optimal methods (AIPW, AIPW-SN, TMLE, C-Learner) in this setting, demonstrating strong numerical stability despite extreme inverse propensity weights. In some cases, the C-Learner improves upon AIPW and TMLE by orders of magnitude.[6]

[6]The median absolute error is reported in [31]. Qualitative results are the same under this additional metric, which we omit for easier exposition. In Section E.1.3, we show similar results for other metrics such

Even when outcomes are not truly bounded (here they are Gaussian), it is common to implement the TMLE via a logistic link for improved estimator stability. Therefore, in addition to TMLE (8) with the squared loss, we also compare with the logistic formulation of TMLE ("TMLE-L") using the `tmle` R package [24]. To use the logistic link, one needs to normalize the values of Y to be between zero and one, which requires an upper and lower bound for Y. Even when Y is not actually bounded, such lower and upper bounds can be roughly estimated from data. We can also formulate the C-Learner using the logistic link, which we call "C-Learner-L". We find that C-Learner-L improves upon TMLE-L as well, demonstrating that C-Learner can be used in conjunction with other heuristics to improve and stabilize estimates. Details of C-Learner with the logistic link are in Section E.

Numerical Instability and Common Ways To Mitigate ATE estimation methods involving inverse propensity weights are known to be numerically unstable if the propensity score $\widehat{\pi}(X)$ is close to 0. One way to address this instability is through normalizing these inverse propensity score weights. We find that self-normalized versions of IPW and AIPW are more stable and have better performance than their original versions, e.g. in Table 1.

Another common heuristic to handle extreme estimated propensity scores is to truncate $\widehat{\pi}(X)$ for a chosen small $\eta > 0$ so that if $\widehat{\pi}(X) < \eta$, we redefine $\widehat{\pi}(X) := \eta$ [19, 17]. As dealing with extreme estimated propensity scores $\widehat{\pi}(X)$ can be challenging, another option is to avoid these difficult X with small $\widehat{\pi}(X)$ entirely by adjusting the estimand to exclude or down-weight values of X for which $\widehat{\pi}(X)$ takes extreme values [18, 33]. We show in Section E.1.3 that while AIPW and TMLE perform poorly using raw estimated propensity scores $\widehat{\pi}$, both perform better after truncation. We emphasize that C-Learner outperforms other asymptotically optimal methods in settings with low overlap, without needing to use heuristics like normalization or truncation.

Lastly, additional assumptions, such as bounded outcomes, can be used to improve estimator stability. We also compare with the logistic formulation of TMLE in Table 1.

---

as RMSE and median absolute error.

Estimator Performance When Varying Overlap Between Treatment and Control In order to understand how quickly estimator performance deteriorates in settings with low overlap, we compare how estimators perform in different data settings that vary by the degree of overlap or variation in propensity scores by modifying the treatment assignment. Specifically, we scale the logit of the treatment mechanism by a parameter c such that if $c = 0$, every observation has an equal chance of being observed, while as c increases, treatment probabilities become more extreme (i.e. closer to 0 or 1). See Section E.1.2 for more details. In Figure 3, we display the empirical density of the the fitted propensity scores for $c \in \{0.25, 1, 1.75\}$. The case in which $c = 1$ is precisely the original setting of Kang and Schafer [31]. In Figure 4, we plot in log scale the mean absolute error of each estimator as a function of the scaling parameter c. The other asymptotically optimal methods and IPW deteriorate quickly as the fitted propensities become extreme, while the C-Learner deteriorates an order of magnitude slower. In Table 5 and Table 6 in Section E.1.2, we compute statistics for learned propensity scores $\widehat{\pi}(X)$ for various values of c, such as the minimum, maximum and standard deviation.

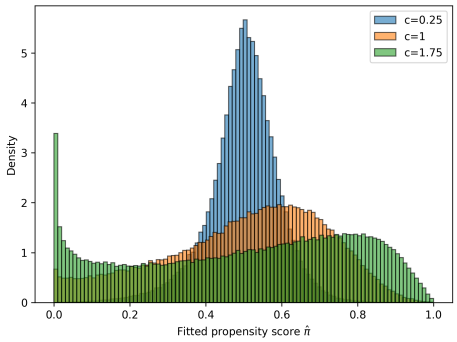

Figure 3. Empirical density of fitted propensity scores $\widehat{\pi}$, for modified Kang and Schafer [31] settings with parameter $c \in \{0.25, 1, 1.75\}$. Histograms across 1000 datasets with size 200 each.

Robins et al. [38] noted that methods like the AIPW, for example, perform poorly in the setting proposed by Kang and Schafer [31] due to high variability in propensity scores. They thus suggest considering the flipped version of the task, in which we want to estimate $\widetilde{\psi}(P) := P[Y(0)]$ instead; there, propensity scores are not extreme, and asymptotically optimal methods such as AIPW work well. Robins et al. [38] thus ask the following question (rewritten to match our notation): "Can we find doubly robust estimators that, under the authors' chosen joint distribution for $(X, A, (1-A)Y)$, both perform almost as well as the direct method for $P[Y(1)]$ and yet perform better than the direct method for $P[Y(0)]$?" We observe that the C-Learner is such an estimator, as it performs well for estimating both $P[Y(0)]$ and $P[Y(1)]$. See Table 10 in Section E.1.3 for $P[Y(0)]$.

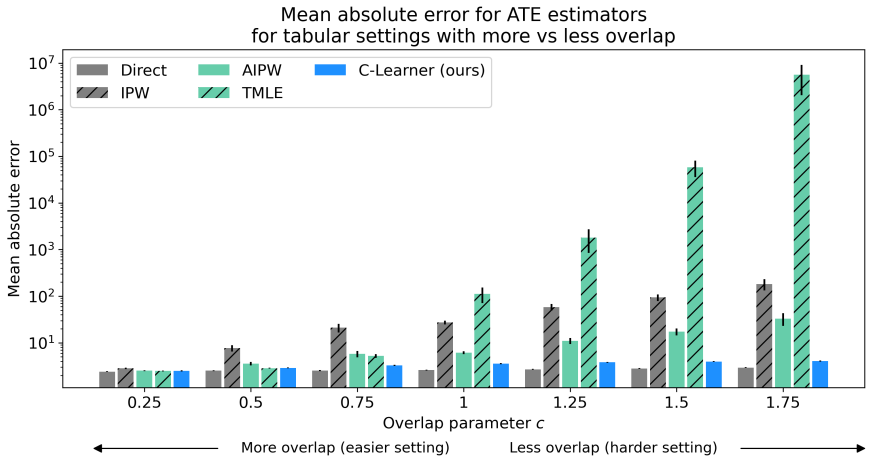

Figure 4. Existing debiased methods (in teal) perform worse in settings with low overlap (larger values of c), in contrast to the simple plug-in estimator ("direct") and C-Learner. Mean average error in the modified Kang and Schafer [31] setting (Section 5.1), for linear outcome models, with different values of scaling parameter c. Results are averaged over 1000 dataset draws for each c, each with $N = 200$. Error bars are ± 1 standard error. This figure depicts the same information as the top row in Figure 2, but with more values of c.

5.1.2 The Dual C-Learner and Covariate Balancing.

Here, we include experiment results in this setting for the dual C-Learner and for balancing methods for the propensity score, as introduced in Section 3.3. We now compare an alternative formulation of the C-Learner presented in Equation (14), in which we take the linear outcome model (learned with no constraint) as fixed and we solve for the best propensity model subject to the efficiency constraints for the propensity weighted estimator. Finally, we use this learned propensity model for inverse propensity weighting to obtain the dual C-Learner estimator. We present in Table 3 a comparison between the following propensity weighted estimators: IPW, which would be analogous to a direct implementation, IPW with the standard CBPS formulation that forces balancing across the covariates X, a parametrically fluctuated model ("Param-Fluc") based on [44], which is analogous to the parametric fluctuation of TMLE, and the dual C-Learner ("C-Learner-Dual") as just described. We also include self-normalized versions of each method (denoted with "-SN"). Implementation details are in Section E. For both sample sizes, the dual C-Learner is the best-performing propensity-weighted estimator. See Section E.1.3 for additional results that use covariate-balanced propensity scores.

	(a) $N = 200$				(b) $N = 1000$			
Method	Bias		Mean Abs Err		Bias		Mean Abs Err	
Direct	-5.12	(0.10)	5.30	(0.09)	-3.48	(0.04)	3.49	(0.04)
IPW	1192	(919)	1206	(919)	28.3	(3.23)	32.30	(3.23)
IPW-SN	-1.01	(0.10)	7.29	(0.33)	2.26	(0.29)	4.85	(0.26)
Lagrangian	-4.44	(0.10)	3.57	(0.09)	-2.29	(0.04)	2.34	(0.05)
AIPW	275	(215)	280	(215)	2.28	(0.52)	4.58	(0.51)
AIPW-SN	**-0.82**	(0.24)	4.53	(0.19)	**0.36**	(0.14)	2.74	(0.15)
TMLE	487	(345)	10927	(493)	17.3	(10.20)	20.05	(10.23)
C-Learner	-2.89	(0.10)	**3.53**	(0.07)	-1.92	(0.04)	**2.03**	(0.04)

Table 2. Comparison of estimators on misspecified Kang and Schafer [31] settings, in 1000 tabular simulations, for gradient boosted regression tree outcome models (Section 5.1.3). Asymptotically optimal methods are listed beneath the horizontal divider. We highlight the best-performing *asymptotically optimal* method in **bold**. Standard errors are displayed in parentheses to the right of the point estimate.

5.1.3 Gradient Boosted Regression Tree Outcome Models.

We now demonstrate the flexibility of the C-Learner by using gradient boosted regression trees as outlined in Section 4.3. As before, the propensity model $\widehat{\pi}$ is fit as a logistic regression on covariates X. Since sample splitting is more appropriate for this flexible model class, we use cross-fitting with $K = 2$ folds as we describe in Section 4.1, and treat more formally in Section 6. Additional implementation details, such as the grid of hyperparameters for tuning and coverage results are deferred to Section E.1.1.

The results for the mean absolute error and their respective standard errors are displayed in Table 2. The C-Learner outperforms the direct method and achieves the best mean absolute error (MAE) among *all* estimators and across *all* sample sizes. For comparison, we also include a plug-in method where the outcome model is learned using only the first stage in Section 4.3. Note that the first stage by itself does not aim to make the estimate of the first-order error term zero, so that the resulting estimator is not asymptotically optimal. In the results in Table 2 this is labeled as "Lagrangian", as it can be seen as a Lagrangian relaxation of the C-Learner framework. Especially for small sample sizes (e.g. $N = 200$), the IPW, AIPW, and TMLE estimators perform very poorly, likely due to being sensitive to extreme propensity weights. Self-normalization for both IPW ("IPW-SN") and AIPW ("AIPW-SN") is crucial in these settings.

	(a) $N = 200$				(b) $N = 1000$			
Method	Bias		Mean Abs Err		Bias		Mean Abs Err	
IPW	22.10	(2.58)	27.28	(2.53)	-0.43	(0.04)	1.17	(0.03)
IPW-SN	3.36	(0.29)	5.42	(0.26)	105.46	(59.84)	105.67	(59.84)
CBPS	1.98	(0.18)	4.30	(0.13)	-1.85	(0.07)	2.36	(0.06)
CBPS-SN	-1.20	(0.10)	2.76	(0.06)	-1.35	(0.04)	1.59	(0.04)
Param-Fluc	-2.77	(0.13)	3.61	(0.11)	-1.74	(0.04)	1.85	(0.04)
Param-Fluc-SN	-1.81	(0.10)	2.79	(0.07)	-1.72	(0.04)	1.83	(0.04)
C-Learner-Dual	**-0.01**	(0.10)	**2.59**	(0.06)	**-0.41**	(0.04)	**1.16**	(0.03)
C-Learner-Dual-SN	-9.20	(0.11)	9.22	(0.11)	-9.49	(0.05)	9.49	(0.05)

Table 3. Comparison of estimator performance on mispecified datasets from Kang and Schafer [31] in 1000 tabular simulations, for linear logit propensity models (Section 5.1.2). We highlight the best-performing method *overall* in **bold**. Standard errors are displayed within parentheses to the right of the point estimate.

In Table 9 in Section E.1.3, we present results when truncating $\widehat{\pi}(X)$ at arbitrary thresholds of of 0.1% and 5%. There, C-Learner again performs better (truncating at 0.1%) or similarly to TMLE and AIPW-SN (truncating at 5%), and better than other methods.

5.2 CivilComments Dataset and Neural Network Language Models

Neural networks can learn good feature representations for image and text data. We construct a new semisynthetic causal inference dataset using text covariates.

Setting Content moderation is a fundamental problem for maintaining the integrity of social media platforms. We consider a setting in which we wish to measure the average level of toxicity across all user comments. It is infeasible to have human experts label all comments for toxicity, and the comments that get flagged for human labeling may be a biased sample. This is a mean missing outcome problem (Section 2.1).

We use the CivilComments dataset [16], which contains real-world online comments and corresponding human-labeled toxicity scores. The dataset contains toxicity labels $Y(1)$ for all comments X (which provides a ground truth to compare to), and we construct the labeling (treatment) mechanism $A \in \{0, 1\}$ to induce selection bias. Specifically, whether the toxicity label for a comment can be observed is drawn as $A \sim \text{Bernoulli}(g(X))$ where $g(X) = \text{clip}(b(X), l, u)$ and $\text{clip}(y, l, u)$ is $\max(l, y)$ if $y \leq l$, and $\min(u, y)$ if $y > u$. We set $u = 0.9$ and $l = 10^{-4}$. A lower l implies more extreme $\pi(X)$ (and thus also $\widehat{\pi}(X)$), i.e.

Method	Bias		Mean Abs Err	
Direct	0.173	(0.008)	0.177	(0.007)
IPW	0.504	(0.084)	0.546	(0.081)
IPW-SN	0.114	(0.017)	0.153	(0.014)
AIPW	0.084	(0.043)	0.307	(0.032)
AIPW-SN	0.116	(0.018)	0.161	(0.014)
TMLE	-1.264	(1.361)	1.802	(1.355)
C-Learner (best val MSE)	0.103	(0.015)	0.141	(0.011)
C-Learner (smallest bias shift)	**0.075**	(0.012)	**0.115**	(0.008)

Table 4. Comparison of estimators in the CivilComments [16] semi-synthetic dataset (Section 5.2) over 100 re-drawn datasets, with $l = 10^{-4}$. Asymptotically optimal methods are listed beneath the horizontal divider. We highlight the best-performing method in **bold**. Estimators (besides the "smallest bias shift" C-Learner) are analogous to those in Section 5.1. Standard errors are displayed within parentheses to the right of the point estimate.

less overlap. Here, $b(X) \in [0, 1]$ is a continuous measure of whether comment X relates to the demographic identity "Black", from the dataset. The labeled data suffers from the following selection bias: within this dataset, comments mentioning the demographic identity "Black" tend to be labeled as more toxic, compared to ones that don't, so a naive average of toxicity over labeled units would overestimate the overall toxicity. From a causal perspective, we have induced confounding, can be handled by ATE estimators.

Procedure We demonstrate C-Learner with neural networks. To learn $\widehat{\pi}$, $\widehat{\mu}$, and $\widehat{\mu}_C$, we fine-tune a pre-trained DistilBERT model [40] with a linear head using stochastic gradient descent, with $\widehat{\mu}^C$ learned using the procedure described in Section 4.4. We train $\widehat{\mu}$ using squared loss on labeled units, and propensity models $\widehat{\pi}$ using the logistic loss. We fit nuisance estimators on 100 re-drawn datasets of size 2000 each, from the full dataset of size 405,130. On each dataset draw, we use cross-fitting, as described in Section 6, with $K = 2$ folds, for all estimators. We consider two ways of picking hyperparameters (λ and learning rate), as described in Section 4.4: one in which we choose hyperparameters for the best mean squared error on validation data, and one in which we choose hyperparameters that minimize the size of the bias shift. Additional details are in Section E.2.1.

Results In Table 4 (a low overlap setting with $l = 10^{-4}$), both variants of C-Learner In Section E.2.2 we also investigate settings with both low and high overlap ($l = 10^{-2}, 10^{-3}, 10^{-4}$). We observe C-Learner performs better compared to other asymptotically optimal methods in datasets with low overlap, and more comparably to other asymptotically optimal

methods with higher overlap. We also display a subset of these results in the second row of Figure 2, with λ chosen to have the best validation MSE.

These results with text covariates echo our tabular results (Section 5.1), suggesting C-Learner's superior performance in low-overlap extends to more complex settings.

6 Asymptotic Properties

We identify conditions under which the C-Learner is semiparametrically efficient and doubly robust. In particular, our theoretical treatment also shows the asymptotic optimality of one-step estimation (self-normalized AIPW) and targeting (TMLE with unbounded real outcomes) since they are also C-Learners satisfying the requisite conditions.

The proof techniques here are common across those used for other first-order de-biasing methods [52, 55, 51, 48, 32] as we share the goal of setting the first-order error term in the distributional Taylor expansion to zero, and also showing second-order terms are negligible.

We focus on the cross-fitted [54, 13] formulation of the C-Learner where we split the data so that nuisance estimators are fitted on a training (auxiliary) fold and evaluated to form a causal estimator on an evaluation (main) fold. Divide the dataset of size n into K disjoint cross-fitting splits. Assuming K evenly divides n for simplicity, let $P_{k,n}$ be the empirical measure over data from the k-th folds, and $P_{-k,n}$ be the empirical measure over data from all other folds. For each $k = 1, \ldots, K$, on the training fold $P_{-k,n}$, we train a propensity score model $\widehat{\pi}_{-k,n}(x)$ to estimate $P[A \mid X = x]$. The C-Learner optimizes the prediction loss evaluated under $P_{-k,n}$, subject to the first-order correction constraint evaluated on the evaluation fold $P_{k,n}$, with final estimator $\widehat{\psi}_n^C$:

$$\widehat{\mu}_{-k,n}^C \in \operatorname*{argmin}_{\tilde{\mu} \in \mathcal{F}} \left\{ P_{-k,n}[A(Y - \tilde{\mu}(X))^2] : P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)} (Y - \tilde{\mu}(X)) \right] = 0 \right\} \tag{22}$$

$$\widehat{\psi}_n^C := \frac{1}{K} \sum_{k=1}^{K} P_{k,n}[\widehat{\mu}_{-k,n}^C(X)]. \tag{23}$$

Using notation from Section 4.1, for fold k, let $P_{\text{train}} = P_{-k,n}$, and $P_{\text{val}} = P_{\text{eval}} = P_{k,n}$.

In addition to its practical benefits, cross-fitting simplifies the proof of asymptotic optimality, especially when nuisance parameter models can be large and complex. When

nuisance model classes are Donsker—converging at $\sqrt{n}$-rates—asymptotic optimality can be shown without explicit sample splitting [32, 46, 59]. In contrast to standard cross-fitting, in which nuisance parameters are learned only on the training split $P_{-k,n}$, the constraint (22) is over $P_{k,n}$. We shortly identify sufficient conditions that guarantee asymptotic optimality (Assumption D), in addition to standard conditions on the nuisance parameters $\widehat{\pi}_{-k,n}, \widehat{\mu}^C_{-k,n}$ (Assumption A, Assumption B) and on outcomes (Assumption C).

Assumption A (Overlap)**.** *For some $\eta > 0$ and for all k, n, we have*

$$\eta \leq \pi(X) \leq 1 - \eta, \quad \eta \leq \widehat{\pi}_{-k,n}(X) \leq 1 - \eta \quad a.s.$$

Assumption B (Convergence rates of propensity and constrained outcome models)**.** *For all $k \in \{1, \ldots, K\}$, both $\widehat{\pi}, \widehat{\mu}^C$ are consistent,*

$$\|\widehat{\pi}_{-k,n} - \pi\|_{L_2(P)} = o_P(1), \quad \|\widehat{\mu}^C_{-k,n} - \mu\|_{L_2(P)} = o_P(1)$$

$$\textit{and also} \quad \|\widehat{\pi}_{-k,n} - \pi\|_{L_2(P)} \cdot \|\widehat{\mu}^C_{-k,n} - \mu\|_{L_2(P)} = o_P(n^{-\frac{1}{2}}).$$

As we discuss below, we can relax these assumptions when guaranteeing double robustness (consistency under misspecified nuisance parameters). As is typical, we assume that outcomes do not differ too much from their means, conditional on covariates.

Assumption C (Outcomes are close to conditional means)**.** *For all k, n, and for some $0 < B < \infty$,*

$$\|A(Y - P[Y \mid X])\|_{L_2(P)} = \|A(Y - \mu(X))\|_{L_2(P)} \leq B.$$

We also require the following condition (Assumption D) on the C-Learner outcome model. This condition is new to our setting, and warrants discussion.

Assumption D (Empirical process assumption)**.**

$$(P_{k,n} - P)(\widehat{\mu}^C_{-k,n}(X) - \mu(X)) = o_P(n^{-1/2}).$$

In Section 3.1, we showed how versions of one-step estimation and targeting can also be considered C-Learners. We show in Section B.4 and Section B.5 that cross-fitted versions

of the self-normalized AIPW and TMLE for continuous and unbounded outcomes, under standard assumptions on $\widehat{\pi}_{-k,n}, \widehat{\mu}_{-k,n}$ (with $\widehat{\mu}_{-k,n}$ the usual outcome model learned using Equation (9)) also satisfy Assumption D. Thus, C-Learner results in Theorems 2, 3 (to come) apply directly to these estimators. One future research direction is to understand the model classes and constrained optimization methods for which this assumption holds.

Assumption D is not implied by previous assumptions. Consider the following example:

Example 1 (Assumption D does not follow from Assumptions A, B, C)**:** Assume Assumption A. Let $\widehat{\mu}_{-k,n}$ be an *unconstrained* solution to $\widehat{\mu}_{-k,n} \in \operatorname{argmin}_{\tilde{\mu}\in\mathcal{F}} P_{-k,n}[A(Y-\tilde{\mu}(X))^2]$, in contrast txo the constrained problem in (22). Then, define $\widehat{\mu}^C_{-k,n}(x) = \widehat{\mu}_{-k,n}(x) + \sum_{i=1}^{n/K} \mathbf{1}\{x = x_i\}$, with the sum of indicators over elements in the $P_{k,n}$ fold. Then

$$P\left|\widehat{\mu}^C_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)\right| = 0 \quad \text{and} \quad P_{k,n}\left(\widehat{\mu}^C_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)\right) = 1. \tag{24}$$

We make the following additional assumptions for this example. We also assume that $\|\widehat{\mu}_{-k,n}(X) - \mu(X)\| = o_P(1)$ (analogous to Assumption B). Let $\widehat{\mu}$ denote $\operatorname{argmin}_{\tilde{\mu}\in\mathcal{F}} P[A(Y-\tilde{\mu}(X))^2]$, the best fitted model over the entire population. For simplicity, assume that Y is bounded to satisfy Assumption C, and that $\widehat{\mu}(X)$ is bounded as well. Then

$$\begin{aligned}(P_{k,n} - P)\left(\widehat{\mu}^C_{-k,n}(X) - \mu(X)\right) &= (P_{k,n} - P)\left(\widehat{\mu}^C_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)\right) \\ &+ (P_{k,n} - P)\left(\widehat{\mu}_{-k,n}(X) - \widehat{\mu}(X)\right) \\ &+ (P_{k,n} - P)\left(\widehat{\mu}(X) - \mu(X)\right)\end{aligned}$$

where the first term on the RHS is 1 from (24), the second is $o_P(1)$ by the cross-fitting lemma (Lemma 1 in Section B.3), and the third is $o_P(1)$ by Chebyshev. Therefore,

$$(P_{k,n} - P)\left(\widehat{\mu}^C_{-k,n}(X) - \mu(X)\right) = 1 + o_P(1)$$

so that Assumption D does not hold for this example, while Assumptions A, B, C do. $\diamond$

The C-Learner (23) enjoys the following asymptotics; see Section B.1 for the proof.

Theorem 2 (Asymptotic variance of C-Learner)**.** *Under Assumptions A, B, C, and D,*

$$\sqrt{n}(\widehat{\psi}^C_n - \psi(P)) \overset{d}{\rightsquigarrow} N(0, \sigma^2) \quad \textit{where} \quad \sigma^2 := \operatorname{Var}_P\left(\frac{A}{\pi(X)}(Y - \mu(X)) + \mu(X)\right).$$

Since σ^2 is the semiparametric efficiency bound for the ATE estimand [45, 32], the C-Learner (23) achieves the tightest confident interval and is optimal in the usual local asymptotic minimax sense [59, Theorem 25.21].

Double robustness By virtue of their first-order correction, standard approaches like one-step estimation and targeting enjoy double robustness: if either of the propensity model or the outcome model is consistent, then the resulting estimator is consistent. We show a similar guarantee for the C-Learner. Here, we assume that either $\widehat{\pi}_{-k,n}$ $\widehat{\mu}^C_{-k,n}$ is consistent.

Assumption E (At least one of $\widehat{\pi}, \widehat{\mu}^C$ is consistent)**.** *For all k, the product of the errors for the outcome and propensity models decays as*

$$\|\widehat{\pi}_{-k,n} - \pi\|_{L_2(P)} \cdot \|\widehat{\mu}^C_{-k,n} - \mu\|_{L_2(P)} = o_P(1).$$

Using Assumption E in place of Assumption B, we arrive at the following result; see Section B.2 for the proof.

Theorem 3 (C-Learner is doubly robust)**.** *The C-Learner (23) is consistent under Assumptions A, C, D, and E.*

"Dual" C-Learner The results in Theorem 2 and Theorem 3 are extended to the "Dual" C-Learner as defined in Equation (14), and are stated and proved in Section D.

7 Discussion

We introduce a constrained learning framework for first-order debiasing in causal estimation and semiparametric inference. We pose asymptotically optimal plug-in estimators as those whose nuisance parameters are solutions to a optimization problem, under the constraint that the first-order error of the plug-in estimator with respect to the nuisance parameter estimate is zero. This perspective encompasses versions of one-step estimation and targeting.

The constrained learning perspective enables a new method (Constrained Learner, a.k.a. C-Learner), which solves this constrained optimization directly while using the entire model class. It outperforms existing asymptotically optimal methods without additional

heuristics or assumptions in settings with low overlap, and performs similarly otherwise. We demonstrate C-Learner's versatility by instantiating it with model classes including linear models, gradient boosted trees, and neural networks, and on datasets with both tabular and text covariates.

Our theoretical analysis is only a small initial step in building a principled understanding of the benefits of the constrained learning framework. One future direction is to investigate which model classes and optimization methods satisfy our theoretical assumptions (e.g. Assumption D). Finally, we hope this work spurs further investigation on how constrained optimization can enable more robust estimators, by proposing better optimization procedures, or extending to additional model classes and estimands beyond Section A.

References

[1] S. Athey and G. Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

[2] S. Athey, G. W. Imbens, and S. Wager. Approximate residual balancing: debiased inference of average treatment effects in high dimensions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(4):597–623, 2018.

[3] S. Athey, J. Tibshirani, and S. Wager. Generalized random forests. 2019.

[4] L. B. Balzer, M. van der Laan, J. Ayieko, M. Kamya, G. Chamie, J. Schwab, D. V. Havlir, and M. L. Petersen. Two-stage tmle to reduce bias and improve efficiency in cluster randomized trials. *Biostatistics*, 24(2):502–517, 2023.

[5] H. Bang and J. M. Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61, 2005. URL https://api.semanticscholar.org/CorpusID:14135922.

[6] D. Benkeser and M. Van Der Laan. The highly adaptive lasso estimator. In *2016 IEEE international conference on data science and advanced analytics (DSAA)*, pages 689–696. IEEE, 2016.

[7] A. F. Bibaut and M. J. van der Laan. Fast rates for empirical risk minimiza-

tion over c\adl\ag functions with bounded sectional variation norm. *arXiv preprint arXiv:1907.09244*, 2019.

[8] P. Bickel, C. A. J. Klaassen, Y. Ritov, and J. Wellner. *Efficient and Adaptive Estimation for Semiparametric Models*. Springer Verlag, 1998.

[9] P. J. Bickel, C. A. Klaassen, P. J. Bickel, Y. Ritov, J. Klaassen, J. A. Wellner, and Y. Ritov. *Efficient and adaptive estimation for semiparametric models*, volume 4. Springer, 1993.

[10] J. Brooks-Gunn, F.-r. Liaw, and P. K. Klebanov. Effects of early intervention on cognitive function of low birth weight preterm infants. *The Journal of pediatrics*, 120 (3):350–359, 1992.

[11] C. Carvalho, A. Feller, J. Murray, S. Woody, and D. Yeager. Assessing treatment effect variation in observational studies: Results from a data challenge. *arXiv:1907.07592 [stat.ME]*, 2019.

[12] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794. ACM, 2016.

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

[14] V. Chernozhukov, W. K. Newey, V. Quintas-Martinez, and V. Syrgkanis. Automatic debiased machine learning via riesz regression. *arXiv preprint arXiv:2104.14737*, 2021.

[15] V. Chernozhukov, W. Newey, V. M. Quintas-Martínez, and V. Syrgkanis. Riesznet and forestriesz: Automatic debiased machine learning with neural nets and random forests. In *International Conference on Machine Learning*, pages 3901–3914. PMLR, 2022.

[16] cjadams, D. Borkan, inversion, J. Sorensen, L. Dixon, L. Vasserman, and nithum. Jigsaw unintended bias in toxicity classification, 2019.

[17] S. R. Cole and M. A. Hernán. Constructing inverse probability weights for marginal structural models. *American journal of epidemiology*, 168(6):656–664, 2008.

[18] R. K. Crump, V. J. Hotz, G. W. Imbens, and O. A. Mitnik. Moving the goalposts: Addressing limited overlap in the estimation of average treatment effects by changing the estimand. Technical report, National Bureau of Economic Research, 2006.

[19] P. Ding. A first course in causal inference, 2023.

[20] L. T. Fernholz. *Von Mises calculus for statistical functionals*, volume 19. Springer Science & Business Media, 2012.

[21] A. Fisher and E. H. Kennedy. Visually communicating and teaching intuition for influence functions, 2019. URL https://arxiv.org/abs/1810.03260.

[22] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[23] S. Gruber and M. van der Laan. An application of collaborative targeted maximum likelihood estimation in causal inference and genomics. *The International Journal of Biostatistics*, 6(1), 2010.

[24] S. Gruber and M. van der Laan. tmle: an r package for targeted maximum likelihood estimation. *Journal of Statistical Software*, 51:1–35, 2012.

[25] P. R. Hahn, J. S. Murray, and C. Carvalho. Bayesian regression tree models for causal inference: regularization, confounding, and heterogeneous effects, 2019.

[26] J. L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

[27] K. Hirano, G. Imbens, and G. Ridder. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189, 2003.

[28] K. Imai and M. Ratkovic. Covariate balancing propensity score. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):243–263, 2014.

[29] C. Ju, J. Schwab, and M. J. van der Laan. On adaptive propensity score truncation

in causal inference. *Statistical methods in medical research*, 28(6):1741–1760, 2019.

[30] C. Ju, R. Wyss, J. M. Franklin, S. Schneeweiss, J. Häggström, and M. J. van der Laan. Collaborative-controlled lasso for constructing propensity score-based estimators in high-dimensional data. *Statistical methods in medical research*, 28(4):1044–1063, 2019.

[31] J. D. Y. Kang and J. L. Schafer. Demystifying Double Robustness: A Comparison of Alternative Strategies for Estimating a Population Mean from Incomplete Data. *Statistical Science*, 22(4):523 – 539, 2007.

[32] E. H. Kennedy. Semiparametric doubly robust targeted double machine learning: a review. *arXiv:2203.06469 [stat.ME]*, 2022.

[33] F. Li, L. E. Thomas, and F. Li. Addressing Extreme Propensity Scores via the Overlap Weights. *American Journal of Epidemiology*, 188(1):250–257, 09 2018.

[34] R. Nabi, N. S. Hejazi, M. J. van der Laan, and D. C. Benkeser. Statistical learning for constrained functional parameters in infinite-dimensional models with applications in fair machine learning. *ArXiv*, abs/2404.09847, 2024. URL https://api.semanticscholar.org/CorpusID:269149113.

[35] W. K. Newey. The asymptotic variance of semiparametric estimators. *Econometrica*, pages 1349–1382, 1994.

[36] M. Oprescu, V. Syrgkanis, and Z. S. Wu. Orthogonal random forest for causal inference. In *International Conference on Machine Learning*, pages 4932–4941. PMLR, 2019.

[37] J. Pfanzagl and W. Wefelmeyer. Contributions to a general asymptotic statistical theory. *Statistics & Risk Modeling*, 3(3-4):379–388, 1985.

[38] J. Robins, M. Sued, Q. Lei-Gomez, and A. Rotnitzky. Comment: Performance of double-robust estimators when" inverse probability" weights are highly variable. *Statistical Science*, 22(4):544–559, 2007.

[39] C. A. Rolling and Y. Yang. Model Selection for Estimating Treatment Effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(4):749–769, 11

2013.

[40] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

[41] C. M. Setodji, D. F. McCaffrey, L. F. Burgette, D. Almirall, and B. A. Griffin. The right tool for the job: choosing between covariate-balancing and generalized boosted model propensity scores. *Epidemiology*, 28(6):802–811, 2017.

[42] C. Shi, D. M. Blei, and V. Veitch. Adapting neural networks for the estimation of treatment effects, 2019.

[43] J. R. K. M. D. C. E. H. Y. W. L. S. V. B. R. B. S. T. *nloptr: R Interface to NLopt Optimization Library*, 2023. URL https://CRAN.R-project.org/package=nloptr. R package version 2.0.3.

[44] Z. Tan. Bounded, efficient and doubly robust estimation with inverse weighting. *Biometrika*, 97(3):661–682, 2010.

[45] A. Tsiatis. *Semiparametric theory and missing data*. Springer Science & Business Media, 2007.

[46] A. v. d. Vaart and J. A. Wellner. Empirical processes. In *Weak Convergence and Empirical Processes: With Applications to Statistics*, pages 127–384. Springer, 2023.

[47] L. van der Laan, M. Carone, A. Luedtke, and M. van der Laan. Adaptive debiased machine learning using data-driven model selection techniques. *arXiv preprint arXiv:2307.12544*, 2023.

[48] M. van der Laan. A generally efficient targeted minimum loss based estimator based on the highly adaptive lasso. *The international journal of biostatistics*, 13(2), 2017.

[49] M. van der Laan. Higher order spline highly adaptive lasso estimators of functional parameters: Pointwise asymptotic normality and uniform convergence rates. *arXiv preprint arXiv:2301.13354*, 2023.

[50] M. van der Laan and S. Gruber. Collaborative double robust targeted maximum

likelihood estimation. *The international journal of biostatistics*, 6(1), 2010.

[51] M. van der Laan and S. Gruber. One-step targeted minimum loss-based estimation based on universal least favorable one-dimensional submodels. *The international journal of biostatistics*, 12(1):351–378, 2016.

[52] M. van der Laan and D. Rubin. Targeted maximum likelihood learning. *The international journal of biostatistics*, 2(1), 2006.

[53] M. van der Laan, S. Rose, J. S. Sekhon, S. Gruber, K. E. Porter, and M. J. van der Laan. Propensity-score-based estimators and c-tmle. *Targeted Learning: Causal Inference for Observational and Experimental Data*, pages 343–364, 2011.

[54] M. van der Laan, S. Rose, W. Zheng, and M. van der Laan. Cross-validated targeted minimum-loss-based estimation. *Targeted learning: causal inference for observational and experimental data*, pages 459–474, 2011.

[55] M. van der Laan, S. Rose, et al. *Targeted learning: causal inference for observational and experimental data*, volume 4. Springer, 2011.

[56] M. van der Laan, S. Qiu, and L. van der Laan. Adaptive-tmle for the average treatment effect based on randomized controlled trial augmented with real-world data. *arXiv preprint arXiv:2405.07186*, 2024.

[57] M. J. Van der Laan and S. Rose. *Targeted learning: causal inference for observational and experimental data*, volume 10. Springer, 2011.

[58] M. J. van der Laan, D. Benkeser, and W. Cai. Efficient estimation of pathwise differentiable target parameters with the undersmoothed highly adaptive lasso. *The International Journal of Biostatistics*, 19(1):261–289, 2023.

[59] A. W. Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[60] S. Wager and S. Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[61] Q. Zhao. Covariate balancing propensity score by tailored loss functions. 2019.

A Extending C-Learner to Other Estimands

We briefly sketch how the C-Learner can be extended to other estimands, beyond just the ATE with $Y(0) = 0$ as in Section 2.1. Let $Z := (W, Y) \sim P$ and let the target functional $\psi(P)$ be continuous and linear in $\mu(W) = P(Y \mid W)$, the conditional distribution of Y given W. For example, we let $W = (X, A)$ in the ATE setting. For other functionals that admit a distributional Taylor expansion with canonical gradient with respect to $P_{Y,W|X}$, $\varphi(Z)$, then we can similarly formulate the C-Learner again as learning the best $\widehat{\mu}$, subject to the constraint that the estimate of the first-order error term is 0.

When $\psi(P)$ is continuous and linear in μ, by the Riesz representation theorem, if $\psi(P)$ is $L_2(P)$-continuous in μ (see for example Equation (4.4) from [35]), then there exists $a \in L_2(P)$ such that for all $\mu \in L_2(P)$,

$$\psi(P) = P[a(W)\mu(W)].$$

This $a(\cdot)$ is commonly referred to as the Riesz representer [14], and the corresponding random variable $a(W)$ can be referred to as the clever covariate [57]. These linear functionals satisfy the following mixed bias property:

$$\psi(\widehat{P}) - \psi(P) + P[\widehat{a}(W)(Y - \widehat{\mu}(W))] = P[(\widehat{a}(W) - a(W))(\widehat{\mu}(W) - \mu(W))],$$

where the first-order term in the distributional Taylor expansion as discussed in Section 2.2 is given by $P[\widehat{a}(W)(Y - \widehat{\mu}(W))]$. We refer the reader to [14] or Proposition 4 of [35] for a discussion about this mixed bias property. Therefore, the C-Learner can be formulated more generally as

$$\widehat{\mu}^C \in \operatorname*{argmin}_{\widetilde{\mu} \in \mathcal{F}} \left\{ P_{\text{train}}[\ell(W, Y; \widetilde{\mu})] : P_{\text{eval}}[\widehat{a}(W)(Y - \widetilde{\mu}(W))] = 0 \right\},$$

where ℓ is an appropriate loss function for the outcome model.

Below, we provide several specific examples demonstrating how C-Learner could be adapted to various target functionals when $W = (X, A)$.

Average Treatment Effect We have seen the mean missing outcome setting (10) for estimating the target functional $\psi(P) = P[\mu(X)]$ where $\mu(x) := P[Y \mid A = 1, X = x]$. The loss function is $\ell(W, Y; \widehat{\mu}) = A(Y - \widehat{\mu}(X))^2$. The Riesz representer is $a(X, A) = A/\pi(X)$.

If we no longer assume that $Y(0) = 0$ and we are interested in estimating the standard average treatment effect

$$\psi(P) = P[Y(1) - Y(0)] = P\left[P[Y \mid A = 1, X]\right] - P\left[P[Y \mid A = 0, X]\right],$$

then the Riesz representer, constrained outcome model, and estimator are, respectively,

$$a(W) = \frac{A}{\pi(X)} - \frac{1 - A}{1 - \pi(X)},$$

$$\widehat{\mu}^C \in \underset{\widetilde{\mu} \in \mathcal{F}}{\operatorname{argmin}} \left\{ P_{\text{train}}[\ell(X, A, Y; \widetilde{\mu})] : P_{\text{eval}}\left[\left(\frac{A}{\widehat{\pi}(X)} - \frac{1 - A}{1 - \widehat{\pi}(X)}\right)(Y - \widetilde{\mu}(X, A))\right] = 0 \right\},$$

$$\widehat{\psi}_{\text{C-Learner}} := P_{\text{eval}}\left[\widehat{\mu}^C(X, 1) - \widehat{\mu}^C(X, 0)\right].$$

Average Policy Effect For off-policy evaluation, the goal is to optimize over assignment policies $c(X) \in \{0, 1\}$ to maximize the expected reward under the policy, using observational data collected under an unknown policy $\pi(x) := P(A = 1 \mid X)$. Assume the usual causal inference assumptions (SUTVA, ignorability, and overlap in Section 2.1). Fixing $c(X)$, the average policy effect is

$$\begin{aligned}
\psi(P) &= P[c(X)Y(1) + (1 - c(X))Y(0)] \\
&= P[c(X)P[Y(1) \mid X] + (1 - c(X))P[Y(0) \mid X]] \\
&= P[c(X)P[Y \mid A = 1, X]] + (1 - c(X))P[Y \mid A = 0, X]]
\end{aligned}$$

Here, the Riesz representer, constrained outcome model, and estimator are, respectively,

$$a(X, A) = c(X)\frac{A}{\pi(X)} + (1 - c(X))\frac{1 - A}{1 - \pi(X)},$$

$$\widehat{\mu}^C \in \underset{\widetilde{\mu} \in \mathcal{F}}{\operatorname{argmin}} \left\{ P_{\text{train}}[\ell(X, A, Y; \widetilde{\mu})] : P_{\text{eval}}\left(c(X)\frac{A}{\pi(X)} + (1 - c(X))\frac{1 - A}{1 - \pi(X)}\right)(Y - \widetilde{\mu}(X, A)) = 0 \right\},$$

$$\widehat{\psi}_{\text{C-Learner}} := P_{\text{eval}}\left[c(X)\widehat{\mu}^C(X,1) + (1-c(X))\widehat{\mu}^C(X,0)\right].$$

B Proofs of Asymptotic Properties

Here, we prove results in Section 6. We show that C-Learner is semiparametrically efficient (Section B.1) and doubly robust (Section B.2), under assumptions in Section 6. We also show that versions of one-step estimation methods (self-normalized AIPW) and targeting methods (TMLE with unbounded continuous outcomes) can satisfy these assumptions (Section B.4, Section B.5), so that semiparametric efficiency and double robustness hold for them immediately as well.

B.1 Proof of Theorem 2

Our proof follows a standard argument. We first use the distributional Taylor expansion (55) to rewrite the estimation error $\widehat{\psi}_n^C - \psi$ as the sum of three terms. Then, we address these terms one by one, and we will show how only one of these terms contributes to asymptotic variance.

Let $Z = (X, A, Y)$ as defined in Section 2.1. Let P denote the true population distribution of Z. Let $\psi(P) = P[P[Y \mid X]]$ as in Section 2.1. Functionals ψ may admit a distributional Taylor expansion, also known as a von Mises expansion [20], where for any distributions $P, \overline{P}$ on Z, we can write

$$\psi(\overline{P}) - \psi(P) = -\int \varphi(z;\overline{P})dP(z) + R_2(\overline{P}, P) \tag{25}$$

where $\varphi(z;P)$ which can be thought of as a "gradient" satisfying the directional derivative formula $\frac{\partial}{\partial t}\psi(P + t(\overline{P} - P))\ |_{t=0} = \int \varphi(z;P)d(\overline{P} - P)(z)$. (W.l.o.g. we assume $\varphi(z;P)$ is centered so that $\int \varphi(z;P)dP(z) = 0$.) Here, $-\int \varphi(z;\overline{P})dP(z)$ is the first-order term, and $R_2(\overline{P}, P)$ is the second-order remainder term, which only depends on products or squares of differences between $P, \overline{P}$.

When ψ is the ATE as in our setting, ψ admits such an expansion [27]

$$\varphi(Z;\overline{P}) := \frac{A}{\overline{\pi}(X)}(Y - \overline{\mu}(X)) + \overline{\mu}(X) - \psi(\overline{P}), \tag{26}$$

where $\overline{\pi}(x) := \overline{P}[A = 1 \mid X]$ and $\overline{\mu}(x) := \overline{P}[Y \mid A = 1, X]$. In particular, we get the following explicit formula for the second-order term

$$R_2(\overline{P}, P) := \int \pi(x) \left(\frac{1}{\overline{\pi}(x)} - \frac{1}{\pi(x)} \right) (\overline{\mu}(x) - \mu(x))\, dP(x). \tag{27}$$

We will apply this to our C-Learner estimator as defined in Section 6. Recall that $\widehat{\pi}_{-k,n}$ is trained to predict treatment A given X using $P_{-k,n}$, and $\widehat{\mu}^C_{-k,n}$ is trained to predict outcome given X and $A = 1$ on $P_{-k,n}$, under the constraint that $P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \widehat{\mu}^C_{-k,n}(X)\right] = 0$. Our C-Learner estimator is the mean of plug-in estimators across folds: for each fold, write $\widehat{\psi}^C_{k,n} = \psi(\widehat{P}^C_{k,n})$ so the C-Learner estimate is the average $\widehat{\psi}^C_n = \frac{1}{K}\sum_{k=1}^K \widehat{\psi}^C_{k,n}$.

Noting that any distribution decomposes $\overline{P} = \overline{P}_X \times \overline{P}_{A|X} \times \overline{P}_{Y|A,X}$ and $\psi(\overline{P}) = \overline{P}_X[\mu(X)]$, the following definitions

$$\begin{aligned} \widehat{P}^C_{X;k,n} &:= P_{X;k,n} \\ \widehat{P}^C_{A|X;k,n}[A = 1 \mid X = x] &:= \widehat{\pi}_{-k,n}(x) \\ \widehat{P}^C_{Y|A,X;k,n}[Y \mid A = 1, X = x] &:= \widehat{\mu}^C_{-k,n}(x) \end{aligned}$$

provide a well-defined joint distribution $\widehat{P}^C_{k,n}$.

For each data fold k, we use the distributional Taylor expansion above, where we replace $\overline{P}$ with the joint distribution $\widehat{P}^C_{k,n}$

$$\begin{aligned} \psi(\widehat{P}^C_{k,n}) - \psi(P) &= -P\varphi(Z; \widehat{P}^C_{k,n}) + R_2(\widehat{P}^C_{k,n}, P) \\ &= (P_{k,n} - P)\varphi(Z; P) - P_{k,n}\varphi(Z; \widehat{P}^C_{k,n}) \\ &\quad + (P_{k,n} - P)(\varphi(Z; \widehat{P}^C_{k,n}) - \varphi(Z; P)) + R_2(\widehat{P}^C_{k,n}, P). \end{aligned} \tag{28}$$

Observe that by using Equation (56) and the definition of the C-Learner,

$$\begin{aligned} P_{k,n}\varphi(Z; \widehat{P}^C_{k,n}) &= P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \widehat{\mu}^C_{-k,n}(X)) + \widehat{\mu}^C_{-k,n}(X) - \psi(\widehat{P}^C_{k,n})\right] \\ &= \underbrace{P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \widehat{\mu}^C_{-k,n}(X))\right]}_{=0 \text{ by C-Learner constraint}} + \underbrace{P_{k,n}[\widehat{\mu}^C_{-k,n}] - P_{k,n}[\widehat{\mu}^C_{-k,n}]}_{=0} = 0. \end{aligned}$$

Taking the average of Equation (58) over $k = 1, \ldots, K$, we can write the error $\widehat{\psi}_n^C - \psi$ as the sum of three terms.

$$\widehat{\psi}_n^C - \psi = \frac{1}{K}\sum_{k=1}^{K} \underbrace{(P_{k,n} - P)\varphi(Z;P)}_{S_k^*} + \frac{1}{K}\sum_{k=1}^{K} \underbrace{(P_{k,n} - P)\left(\varphi(Z;\widehat{P}_{k,n}^C) - \varphi(Z;P)\right)}_{T_{1k}} + \frac{1}{K}\sum_{k=1}^{K} \underbrace{R_2(\widehat{P}_{k,n}^C, P)}_{T_{2k}}. \tag{29}$$

Using the decomposition (59), we write

$$S^* = \frac{1}{K}\sum_{k=1}^{K} S_k^*, \quad T_1 = \frac{1}{K}\sum_{k=1}^{K} T_{1k}, \quad T_2 = \frac{1}{K}\sum_{k=1}^{K} T_{2k}, \tag{30}$$

so that $\widehat{\psi}_n^C - \psi = S^* + T_1 + T_2$. We address the terms S^*, T_1 and T_2 separately. The first term can be rewritten as

$$S^* = \frac{1}{K}\sum_{i=1}^{K}(P_{k,n} - P)\varphi(Z;P) = (P_n - P)\varphi(Z;P)$$

so that by the central limit theorem,

$$\sqrt{n}S^* \overset{d}{\rightsquigarrow} N\left(0, \mathrm{Var}_P(\varphi(Z;P))\right).$$

Observe that this quantity depends only on ψ and P, so that it cannot be made smaller by choice of estimator. If the variance of an estimator for ψ is $\mathrm{Var}_P(\varphi(Z;P))$, then it is semiparametrically efficient in the local asymptotic minimax sense (Theorem 25.21 of [59]). Thus, it suffices to show that the rest of the terms, T_1 and T_2, are $o_P(n^{-1/2})$, so that

$$\sqrt{n}(\widehat{\psi}_n^C - \psi) = \sqrt{n}S^* + o_P(1) \overset{d}{\rightsquigarrow} N\left(0, \mathrm{Var}_P(\varphi(Z;P))\right).$$

For a fixed k, $|T_{1k}| = o_P(n^{-1/2})$ by Assumption D so that $|T_1| = o_P(n^{-1/2})$ as desired. The second-order remainder term in the distributional Taylor expansion (55) where we replace

$\overline{P}$ with $\widehat{P}^C_{k,n}$ is

$$T_{2k} = \int \pi(x) \left(\frac{1}{\widehat{\pi}_{-k,n}(x)} - \frac{1}{\pi(x)} \right) \left(\widehat{\mu}^C_{-k,n}(x) - \mu(x) \right) \, dP(x).$$

Under the overlap assumption (Assumption A) and Cauchy-Schwarz,

$$\begin{aligned} |T_{2k}| &\leq \frac{1}{\eta} \int |\widehat{\pi}_{-k,n}(x) - \pi(x)| \left| \widehat{\mu}^C_{-k,n}(x) - \mu(x) \right| \, dP(x) \\ &\leq \frac{1}{\eta} \left\| \widehat{\pi}_{-k,n} - \pi \right\|_{L_2(P)} \left\| \widehat{\mu}^C_{-k,n} - \mu \right\|_{L_2(P)}. \end{aligned} \tag{31}$$

By Assumption B, $|T_{2k}| = o_P(n^{-1/2})$ so that $|T_2| = o_P(n^{-1/2})$ as desired.

B.2 Proof of Theorem 3

We briefly sketch the proof as it is a minor modification of our previous proof in Section B.1. To show the C-Learner estimator $\widehat{\psi}^C_n$ is consistent (rather than that our estimator has the desired asymptotics) under Assumption E (rather than Assumption B), we again rewrite the error $\widehat{\psi}^C_n - \psi$ as a sum of three terms and show each converges to 0 in probability:

- S^*: This term converges to 0 in probability.
- T_{2k}: This also converges to 0 in probability by the same logic as before. Note we only need this to be $o_P(1)$ and not $o_P(n^{-1/2})$ as we would have required for efficiency.
- T_{1k}: This converges to 0 in probability by Assumption D.

B.3 Showing Self-Normalized AIPW and TMLE Satisfy C-Learner Conditions for Theorems 2 and 3

As self-normalized AIPW and TMLE involve simple adjustments to the unconstrained outcome models, in this section, we state assumptions on the *unconstrained* outcome model $\widehat{\mu}_{-k,n}$, fitted in the usual manner on the auxiliary fold $P_{-k,n}$

$$\widehat{\mu}_{-k,n} \in \underset{\widetilde{\mu} \in \mathcal{F}}{\operatorname{argmin}} \, P_{-k,n}[A(Y - \widetilde{\mu}(X))^2],$$

and also assumptions on the gap between the constrained and unconstrained outcome models. Also in this section, we show how these aforementioned assumptions satisfy the C-Learner assumptions required for Theorems 2 and 3.

Then in the following sections, we show how self-normalized AIPW and TMLE satisfy the assumptions stated in this section on the gap between the constrained and unconstrained outcome models.

Unconstrained Outcome Models The following assumption on unconstrained outcome models is analogous to Assumption B. This assumption is standard.

Assumption F (Convergence rates of propensity and *unconstrained* outcome models)**.** *For all* $k = 1, \ldots, K$*,*

$$\|\widehat{\pi}_{-k,n} - \pi\|_{L_2(P)} = o_P(n^{-\frac{1}{4}}), \quad \|\widehat{\mu}^C_{-k,n} - \mu\|_{L_2(P)} = o_P(n^{-\frac{1}{4}}).$$

Distance Between Constrained and Unconstrained Outcome Models These assumptions essentially ensure that the first-order constraint (22) does not change the outcome model too much asymptotically, i.e., $\widehat{\mu}^C_{-k,n}$ and $\widehat{\mu}_{-k,n}$ are similar asymptotically. This first constraint is used to show Assumption B:

Assumption G (The constraint is negligible asymptotically)**.** *For all* $k = 1, \ldots, K$*,*

$$\|\widehat{\mu}^C_{-k,n} - \widehat{\mu}_{-k,n}\|_{L_2(P)} = o_P(n^{-1/4}).$$

Notably, the distance between the constrained solution $\widehat{\mu}^C_{-k,n}$ and its unconstrained counterpart $\widehat{\mu}_{-k,n}$ can be as large as that between $\widehat{\mu}_{-k,n}$ and the true parameter μ, asymptotically.

This next assumption is used to show Assumption D:

Assumption H (Empirical process assumption on $\widehat{\mu}^C_{-k,n}$ vs $\widehat{\mu}_{-k,n}$)**.**

$$(P_{k,n} - P)(\widehat{\mu}^C_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)) = o_P(n^{-1/2}).$$

See Section C.1 for an example of why this assumption is necessary. Given these assumptions, we show the C-Learner assumptions hold.

Proposition 4 (C-Learner conditions hold, given assumptions on unconstrained outcome models and the gap between constrained and unconstrained models)**.**

Assume Assumptions A, C, F, and H. Then Assumptions A, C, B, D are satisfied.

To show this proposition, it suffices to show Assumptions B and D. We will show these assumptions in the rest of this section.

Showing Assumption B By the triangle inequality,

$$\|\widehat{\mu}^C_{-k,n} - \mu\|_{L_2(P)} \leq \|\widehat{\mu}_{-k,n} - \mu\|_{L_2(P)} + \|\widehat{\mu}^C_{-k,n} - \widehat{\mu}_{-k,n}\|_{L_2(P)}. \tag{32}$$

Showing Assumption D Applying the triangle inequality again yields

$$\begin{aligned} |T_{1k}| &:= \left|(P_{k,n} - P)\left(\varphi(Z;\widehat{P}^C_{k,n}) - \varphi(Z;P)\right)\right| \\ &\leq \left|(P_{k,n} - P)\left(\varphi(Z;\widehat{P}^C_{k,n}) - \varphi(Z;\widehat{P}_{-k,n})\right)\right| + \left|(P_{k,n} - P)\left(\varphi(Z;\widehat{P}_{-k,n}) - \varphi(Z;P)\right)\right| \end{aligned} \tag{33}$$

where $\widehat{P}_{k,n}$ is defined as

$$\begin{aligned} \widehat{P}_{X;k,n} &:= P_{X;k,n} \\ \widehat{P}_{A|X;k,n}[A = 1 \mid X = x] &:= \widehat{\pi}_{-k,n}(x) \\ \widehat{P}_{Y|A,X;k,n}[Y \mid A = 1, X = x] &:= \widehat{\mu}_{-k,n}(x) \end{aligned}$$

with $\widehat{\pi}_{-k,n}, \widehat{\mu}_{-k,n}$ as defined in Section 6, and $\widehat{P}^C_{k,n}$ is defined as in Section 6. The second term in (33) is addressed using a standard argument for cross-fitting, as $P_{k,n}$ and $\widehat{P}_{-k,n}$ (and therefore $\varphi(Z;\widehat{P}_{-k,n})$) use disjoint data. In contrast, the first term is not handled by standard cross-fitting arguments: $\widehat{P}_{-k,n}$ only uses data from all but the k-th fold, while $\widehat{P}^C_{k,n}$ uses the k-th fold, except that $\widehat{\mu}^C_{-k,n}$ is made to satisfy a constraint that *does* use the k-th fold, as described in Section 6. We begin by addressing the second term, which is more standard.

Lemma 1 (Cross-fitting lemma)**.** *Let $\widehat{f}(z)$ be a function estimated from an iid sample $Z^N = (Z_{n+1}, \ldots, Z_N)$, and let P_n denote the empirical measure over $(Z_1, \ldots, Z_n)$, which*

is independent of Z^N. Let f be the function estimated from the full distribution P. Then (omitting arguments for brevity) $(P_n - P)(\widehat{f} - f)$ has zero P-expectation, and P-variance upper bounded by $\frac{1}{n}\|\widehat{f} - f\|^2_{L_2(P)}$.

Proof First note that the conditional mean is 0, i.e. $P\left[(P_n - P)(\widehat{f} - f) \mid Z^N\right] = 0$ since

$$P\left[P_n(\widehat{f} - f) \mid Z^N\right] = P\left(\widehat{f} - f \mid Z^N\right) = P(\widehat{f} - f).$$

The conditional variance is

$$\begin{aligned}\mathrm{Var}_P\left\{(P_n - P)(\widehat{f} - f) \mid Z^N\right\} &= \mathrm{Var}_P\left\{P_n(\widehat{f} - f) \mid Z^N\right\} \\ &= \frac{1}{n}\mathrm{Var}_P\left(\widehat{f} - f \mid Z^N\right) \\ &\leq \frac{1}{n}\|\widehat{f} - f\|^2_{L_2(P)}.\end{aligned}$$

Then for (unconditional) mean and variance, $P\left[(P_n - P)(\widehat{f} - f)\right] = 0$ and

$$\begin{aligned}&\mathrm{Var}_P\left\{(P_n - P)(\widehat{f} - f)\right\} \\ &= \mathrm{Var}_P\left\{P\left[(P_n - P)(\widehat{f} - f) \mid Z^N\right]\right\} + P\left[\mathrm{Var}_P\left\{(P_n - P)(\widehat{f} - f) \mid Z^N\right\}\right] \\ &\leq 0 + \frac{1}{n}\|\widehat{f} - f\|^2_{L_2(P)}.\end{aligned}$$

□

Now we show the second term in the RHS in Equation (33) is $o_P(n^{-1/2})$. To do this, write

$$\varphi(Z; P) = \frac{A}{\pi(X)}(Y - \mu(X)) + \mu(X) - \psi(P) \tag{34}$$

$$\varphi(Z; \widehat{P}_{-k,n}) = \frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \widehat{\mu}_{-k,n}(X)) + \widehat{\mu}_{-k,n}(X) - \psi(\widehat{P}_{-k,n}). \tag{35}$$

Observe that $(P_{k,n} - P)\psi(P) = (P_{k,n} - P)\psi(\widehat{P}_{-k,n}) = 0$ as $\psi(P), \psi(\widehat{P}_{-k,n})$ are constants. Thus, it remains to show that for a fixed k, $(P_{k,n} - P)(\widehat{f}_{k,n} - f) = o_P(n^{-1/2})$, where we

omit arguments for brevity and let

$$\widehat{f}_{k,n} = \frac{A}{\widehat{\pi}_{-k,n}}(Y - \widehat{\mu}_{-k,n}) + \mu_{-k,n}, \tag{36}$$

$$f = \frac{A}{\pi}(Y - \mu) + \mu. \tag{37}$$

We do this by using Lemma 1. Observe that

$$\widehat{f}_{k,n} - f = \left(1 + \frac{A}{\pi}\right)(\mu - \widehat{\mu}_{-k,n}) + \frac{A}{\widehat{\pi}_{-k,n} \cdot \pi}(Y - \widehat{\mu}_{-k,n})(\pi - \widehat{\pi}_{-k,n}). \tag{38}$$

Then using Assumption A,

$$\|\widehat{f}_{k,n} - f\|_{L_2(P)} \leq \left(1 + \frac{1}{\eta}\right)\|\widehat{\mu}_{-k,n} - \mu\|_{L_2(P)} + \frac{1}{\eta^2}\|A(Y - \widehat{\mu}_{-k,n})\|_{L_2(P)}\|\pi - \widehat{\pi}_{-k,n}\|_{L_2(P)}. \tag{39}$$

The leftmost term on the RHS converges to 0, using Assumption F. Note that we can bound the rightmost term using the triangle inequality

$$\left(1 + \frac{1}{\eta^2}\right)\left(\|A(Y - \mu)\|_{L_2(P)} + \|\mu - \widehat{\mu}_{-k,n}\|_{L_2(P)}\right)\|\pi - \widehat{\pi}_{-k,n}\|_{L_2(P)}, \tag{40}$$

which also converges to 0 by Assumption F and Assumption C. Thus combining with Lemma 1 we obtain that the second term on the RHS of Equation (33) is $o_P(n^{-1/2})$, as $n^{1/2}(P_{k,n} - P)(\widehat{f}_{k,n} - f)$ has mean 0 and variance $\leq \|\widehat{f}_{k,n} - f\|_{L_2(P)}$ which converges to 0 as $n \to \infty$.

Now we address the remaining term: we show the first term on the RHS of Equation (33) is $o_P(n^{-1/2})$. We use a similar argument to before: let $\widehat{f}_{k,n}$ be as before, and $\widehat{f}^C_{k,n}$ as below:

$$\widehat{f}_{k,n} := \frac{A}{\widehat{\pi}_{-k,n}}(Y - \widehat{\mu}_{-k,n}) + \widehat{\mu}_{-k,n} \tag{41}$$

$$\widehat{f}^C_{k,n} := \frac{A}{\widehat{\pi}_{-k,n}}(Y - \widehat{\mu}^C_{-k,n}) + \widehat{\mu}^C_{-k,n} \tag{42}$$

where we again omit the $\psi(\widehat{P}^C_{-k,n}), \psi(\widehat{P}_{-k,n})$ terms in $\varphi(\widehat{P}^C_{k,n}), \varphi(\widehat{P}_{k,n})$ from $\widehat{f}^C_{k,n}, \widehat{f}_{k,n}$ above as they are constants. We can't use Lemma 1 since $\widehat{f}^C_{k,n}$ also uses $P_{k,n}$ to fit, so instead, by

using Assumptions A and then H,

$$(P_{k,n} - P)\left(\varphi(Z; \widehat{P}^C_{k,n}) - \varphi(Z; \widehat{P}_{k,n})\right) = (P_{k,n} - P)\left(\widehat{f}^C_{k,n} - \widehat{f}_{k,n}\right) \tag{43}$$

$$\leq \left(1 + \frac{1}{\eta}\right)(P_{k,n} - P)(\widehat{\mu}^C_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)) \tag{44}$$

$$= o_P(n^{-1/2}). \tag{45}$$

B.4 Self-Normalized AIPW Satisfies Assumptions G and H

Recall that we showed how self-normalized AIPW also satisfies the C-Learner formulation in Section 3.1. Here we show that Assumptions G and H hold for self-normalized AIPW, so that Theorems 2 and 3 follow through Proposition 4.

As in the discussion in Section 3.1, self-normalized AIPW is equivalent to the specific C-Learner $\widehat{\mu}^C_{-k,n}$ that is defined by adjusting $\widehat{\mu}_{-k,n}$ by an additive constant:

$$\widehat{\mu}^C_{-k,n}(x) = \widehat{\mu}_{-k,n}(x) + c_{k,n} \quad \text{where} \quad c_{k,n} := \frac{P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \widehat{\mu}_{-k,n}(X))\right]}{P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}\right]}.$$

Showing Assumption G: Since $\widehat{\mu}^C_{-k,n}$ is just a constant offset from $\widehat{\mu}_{-k,n}$, it suffices to show that $c_{k,n} = o_P(n^{-1/4})$, for a fixed k.

First, we address the denominator of $c_{k,n}$. Let $c^{\text{den}}_{k,n} := P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}\right]$ and we will show $1/c^{\text{den}}_{k,n} = O_P(1)$. First note $c^{\text{den}}_{k,n} = P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}\right] \geq P_{k,n}[A]$. Then observe that $P_{k,n}[A] \xrightarrow{p} P[A]$, and that $1/P_{k,n}[A] < \infty$ a.s. for large enough n by Borel-Cantelli lemma (as the probability of the event that $P_{k,n}[A] = 0$ is finitely-summable, as $P(A = 0) < 1$ by the overlap assumption (Assumption A). Then $1/P_{k,n}[A] \xrightarrow{p} 1/P[A]$ for large enough n, so that $1/P_{k,n}[A] = O_P(1)$.

Now we address the numerator of $c_{k,n}$. For brevity, call this $c^{\text{num}}_{k,n}$.

$$c^{\text{num}}_{k,n} = P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \widehat{\mu}_{-k,n}(X))\right] \tag{46}$$

$$= P\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \widehat{\mu}_{-k,n}(X))\right] + (P_{k,n} - P)\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \widehat{\mu}_{-k,n}(X))\right] \tag{47}$$

$$= \underbrace{P\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y-\mu(X))\right]}_{=0} + P\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(\mu(X)-\widehat{\mu}_{-k,n}(X))\right] \tag{48}$$

$$+ (P_{k,n}-P)\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y-\widehat{\mu}_{-k,n}(X))\right]$$

$$= P\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(\mu(X)-\widehat{\mu}_{-k,n}(X))\right] + (P_{k,n}-P)\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y-\widehat{\mu}_{-k,n}(X))\right] \tag{49}$$

so that

$$|c_{k,n}^{\text{num}}| \leq \frac{1}{\eta} P\,|\mu(X)-\widehat{\mu}_{-k,n}(X)| + \frac{1}{\eta}(P_{k,n}-P)\,|Y-\widehat{\mu}_{-k,n}(X)| \tag{50}$$

where the inequality is by the overlap assumption (Assumption A) as $A/\widehat{\pi}_{-k,n}(X) \leq 1/\eta$. We show the terms in the last line are all $o_P(n^{-1/4})$. The first term is upper bounded by $\frac{1}{\eta}\|\widehat{\mu}_{-k,n}(X)-\mu(X)\|_{L_2(P)} = o_P(n^{-1/4})$ by Assumption F. For the second term, we use Lemma 1 to show $n^{1/4}(P_{k,n}-P)|Y-\widehat{\mu}_{-k,n}(X)| \xrightarrow{p} 0$: it has mean of 0 and variance $\leq \frac{n^{1/2}}{n}\|Y-\widehat{\mu}_{-k,n}\|_{L_2(P)}$. This upper bound on variance goes to 0, which follows from Assumptions F and C:

$$\begin{aligned}
\|Y-\widehat{\mu}_{-k,n}\|_{L_2(P)} &\leq \|Y-\mu\|_{L_2(P)} + \|A(\mu-\widehat{\mu}_{-k,n})\|_{L_2(P)} \\
&\leq B + \|\mu-\widehat{\mu}_{-k,n}\|_{L_2(P)} \\
&= B + o_P(n^{-1/4}).
\end{aligned}$$

We arrive at the desired result $c_{k,n} = o_P(n^{-1/4})$ as $c_{k,n} = c_{k,n}^{\text{num}}/c_{k,n}^{\text{den}}$, and we just showed that $1/c_{k,n}^{\text{den}} = O_P(1)$ and $c_{k,n}^{\text{num}} = o_P(n^{-1/4})$. Note that we have shown Assumption G for any sequence of $\widehat{\pi}_{-k,n}$'s, as they are bounded by a constant.

Showing Assumption H: To show $(P_{k,n}-P)(\widehat{\mu}_{-k,n}^{C}(X)-\widehat{\mu}_{-k,n}(X)) = o_P(n^{-1/2})$, in the case of the constant shift $\widehat{\mu}^{C}(X) = \widehat{\mu}(X) + c_{k,n}$, note that $\widehat{\mu}_{-k,n}^{C}(X) - \widehat{\mu}_{-k,n}(X) = c_{k,n}$ for every X. Therefore,

$$(P_{k,n}-P)(\widehat{\mu}_{-k,n}^{C}(X)-\widehat{\mu}_{-k,n}(X)) = (P_{k,n}-P)c_{k,n} = 0,$$

with $c_{k,n} < \infty$ a.s. for large enough n (which exists by Borel-Cantelli lemma, as in when we showed Assumption G).

B.5 TMLE Satisfies Assumptions G and H

Recall that we showed how a version of the TMLE for estimating the ATE with continuous unbounded outcomes also satisfies the C-Learner formulation in Section 3.1. Here we show that a cross-fitted version of the TMLE for estimating the ATE with continuous unbounded outcomes additionally satisfies Assumptions G and H, so that Theorems 2 and 3 follow through Proposition 4. Note that the formulation below fits a separate $\epsilon^\star_{k,n}$ per cross-fitting split for consistency with C-Learner, rather than one $\epsilon^\star$ overall as described in the cross-validated version of TMLE in [57].

$$\widehat{\mu}^C_{-k,n}(X) = \widehat{\mu}_{-k,n}(X) + \epsilon^\star_{k,n} \frac{A}{\widehat{\pi}_{-k,n}(X)}, \quad \text{where } \epsilon^\star_{k,n} = \frac{P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \widehat{\mu}_{-k,n}(X))\right]}{P_{k,n}\left[\frac{A}{\widehat{\pi}^2_{-k,n}(X)}\right]}.$$

Showing Assumption G: Observe that by Assumption A

$$\left\| \epsilon^\star_{k,n} \frac{A}{\widehat{\pi}_{-k,n}(X)} \right\|_{L_2(P)} \leq \frac{1}{\eta} \|\epsilon^\star_{k,n}\|_{L_2(P)}$$

so it suffices to show that $\epsilon^\star_{k,n} = o_P(n^{-1/4})$. Note that since $\widehat{\pi}_{-k,n}(x) \leq 1$ for all x,

$$|\epsilon^\star_{k,n}| \leq \frac{P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \widehat{\mu}_{-k,n}(X))\right]}{P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}\right]} \leq |c_{k,n}| \tag{51}$$

where $c_{k,n}$ is the constant adjustment in Section B.4, so that $\epsilon^\star_{k,n} = o_P(n^{-1/4})$.

Showing Assumption H: We want to show

$$(P_{k,n} - P)(\widehat{\mu}^\star_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)) = (P_{k,n} - P)\left(\epsilon^\star_{k,n} \frac{A}{\widehat{\pi}_{-k,n}(X)}\right) = o_P(n^{-1/2}).$$

Note that

$$(P_{k,n} - P)\left(\epsilon^{\star}_{k,n}\frac{A}{\widehat{\pi}_{-k,n}(X)}\right) = \epsilon^{\star}_{k,n}(P_{k,n} - P)\left(\frac{A}{\widehat{\pi}_{-k,n}(X)}\right)$$

and that $(P_{k,n} - P)\left(\frac{A}{\widehat{\pi}_{-k,n}(X)}\right) = O_P(n^{-1/2})$ by Lemma 1, since $|A/\widehat{\pi}_{-k,n}(X)| \leq 1/\eta$ a.s. by Assumption A. Additionally, $\epsilon^{\star}_{k,n} = o_P(n^{-1/4})$, by the argument above where we showed Assumption G. The desired result follows by taking the product of the rates for $\epsilon^{\star}_{k,n}$ and $(P_{k,n} - P)\left(\frac{A}{\widehat{\pi}_{-k,n}(X)}\right)$.

C Additional Proof Notes

C.1 Why we have Assumption H

In settings where $\widehat{\mu}^{C}_{-k,n}$ is independent of $P_{k,n}$, Assumption G can be used to show Assumption H, e.g using Lemma 1. As a reminder, these assumptions are as follows:

- Repeat of Assumption G: For all k,

$$\|\widehat{\mu}^{C}_{-k,n} - \widehat{\mu}_{-k,n}\|_{L_2(P)} = o_P(n^{-1/4}).$$

- Repeat of Assumption H:

$$(P_{k,n} - P)(\widehat{\mu}^{C}_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)) = o_P(n^{-1/2}).$$

However, Assumption G may not imply Assumption H in our setting, as $\widehat{\mu}^{C}_{-k,n}$ is constructed to depend on $P_{k,n}$. If we allow $\widehat{\mu}^{C}_{-k,n}$ to be related to $\widehat{\mu}_{-k,n}$ in an arbitrary way that follows Assumption G, we can construct $\widehat{\mu}^{C}_{-k,n}, \widehat{\mu}_{-k,n}$ where Assumption H does not hold. Here is an example:

Let $\widehat{\mu}_{-k,n}$ be whatever it would be (it doesn't matter because of how we'll define $\widehat{\mu}^{C}_{-k,n}$ in terms of $\widehat{\mu}_{-k,n}$), and let $P_{k,n}$ consist of $x_1, \ldots, x_{n/K}$. Recall that $\widehat{\mu}^{C}_{-k,n}$ can depend on $x_1, \ldots, x_{n/K}$. Here, we define

$$\widehat{\mu}^{C}_{-k,n}(x) = \widehat{\mu}_{-k,n}(x) + \sum_{i=1}^{n/K} \mathbf{1}\{x = x_i\}$$

so that $\widehat{\mu}^C_{-k,n} = \widehat{\mu}_{-k,n} + \sum_{i=1}^{n/K} \delta(x_i)$ with δ the dirac delta function. Then

$$P|\widehat{\mu}^C_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)| = 0$$

$$P_{k,n}|\widehat{\mu}^C_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)| = 1$$

so that

$$(P_{k,n} - P)|\widehat{\mu}^C_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)| = 1$$

so that Assumption H does not hold. However, Assumption G still holds as

$$P|\widehat{\mu}^C_{-k,n}(X) - \widehat{\mu}_{-k,n}(X)|^2 = 0.$$

Even though $\widehat{\mu}^C_{-k,n}$ is some weird measure 0 modification of $\widehat{\mu}_{-k,n}$, we can also construct other, less weird modifications: for example, instead of adding dirac deltas $\delta(x_i)$, we could instead add kernels of height 1 around $x_1, \ldots, x_{n/K}$ and decreasing width (while adjusting kernels to make sure kernels for different x_i, x_j do not overlap) to again satisfy Assumption G but not Assumption H.

D Proofs of Asymptotic Properties for Dual C-Learner

In this section we formally define the dual C-Learner and show that it is semiparametrically efficient (Section D.1) and doubly robust (Section D.2).

First, we define the dual C-Learner. We use cross-fitting, as in Section 6. Consider K even data splits in a dataset of size n. For each $k = 1, \ldots, K$, on the training fold $P_{-k,n}$, we train an outcome model $\widehat{\mu}_{-k,n}(x)$ to estimate $P[Y \mid X = x, A = 1]$. The C-Learner optimizes the prediction loss evaluated under $P_{-k,n}$, subject to the first-order correction constraint evaluated on the evaluation fold $P_{k,n}$:

$$\widehat{\pi}^{C'}_{-k,n} \in \underset{\tilde{\pi} \in \mathcal{F}_\pi}{\operatorname{argmin}} \left\{ P_{-k,n}[\tilde{\pi}(X)^A (1 - \tilde{\pi}(X))^{1-A}] : P_{k,n} \left[n \sum_{i=1}^{n} \left(1 - \frac{A_i}{\tilde{\pi}(X_i)} \right) \widehat{\mu}_{-k,n} \right] (X_i) = 0 \right\} \tag{52}$$

The final dual C-Learner estimator is given by

$$\widehat{\psi}_n^{C'} := \frac{1}{K}\sum_{k=1}^{K} P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}^{C'}(X)}Y\right]. \tag{53}$$

In this section we show the counterpart of the asymptotic properties for the C-Learner, but for the dual C-Learner. As before, we use cross-fitting with K folds. For brevity we write $\widehat{\psi}_n^{C'}$ to denote the cross-fitted dual C-Learner estimator.

D.1 Asymptotic variance of dual C-Learner

We require the following additional assumptions to show asymptotic variance for the dual C-Learner. Assumptions I and J are the dual versions of Assumptions B and D.

Assumption I (Convergence rates of propensity and constrained outcome models)**.** *For all $k \in \{1, \dots, K\}$, both $\widehat{\pi}^{C'}, \widehat{\mu}$ are consistent,*

$$\|\widehat{\pi}_{-k,n}^{C'} - \pi\|_{L_2(P)} = o_P(1), \quad \|\widehat{\mu}_{-k,n} - \mu\|_{L_2(P)} = o_P(1)$$

and also

$$\|\widehat{\pi}_{-k,n}^{C'} - \pi\|_{L_2(P)} \cdot \|\widehat{\mu}_{-k,n}^{C} - \mu\|_{L_2(P)} = o_P(n^{-\frac{1}{2}}).$$

Assumption J (Empirical process assumption)**.**

$$(P_{k,n} - P)(YA \cdot (\widehat{\pi}_{-k,n}^{C'}(X)^{-1} - \pi(X)^{-1})) = o_P(n^{-1/2}).$$

Like Assumption D, Assumption I may need to be shown on a case-by-case basis for different settings and model classes. Under these assumptions, the dual C-Learner has the following asymptotics:

Theorem 5 (Asymptotic variance of dual C-Learner)**.** *Under Assumptions A, I, C, and J,*

$$\sqrt{n}(\widehat{\psi}_n^{C'} - \psi(P)) \overset{d}{\rightsquigarrow} N(0, \sigma^2) \quad \textit{where} \quad \sigma^2 := \mathrm{Var}_P\left(\frac{A}{\pi(X)}(Y - \mu(X)) + \mu(X)\right).$$

The proof of this theorem is very similar to the proof for Theorem 2 with proof in Section B.1.

Proof Let $Z = (X, A, Y)$ as defined in Section 2.1. Let P denote the true population distribution of Z. Let ψ be the ATE as before, but this time, write

$$\psi(P) = P\left[\frac{A}{P(A=1 \mid X)}Y\right]. \tag{54}$$

Note that this ψ is equivalent to ψ as defined in Section 2.1:

$$P\left[\frac{A}{P(A=1 \mid X)}Y\right] = P\left[\frac{\mathbf{1}(A=1)}{P(A=1 \mid X)}P(Y \mid X)\right] = P[P[Y \mid A=1, X]].$$

Functionals ψ may admit a distributional Taylor expansion, also known as a von Mises expansion [20], where for any distributions $P, \overline{P}$ on Z, we can write

$$\psi(\overline{P}) - \psi(P) = -\int \varphi(z; \overline{P}) dP(z) + R_2(\overline{P}, P) \tag{55}$$

where $\varphi(z; P)$ which can be thought of as a "gradient" satisfying the directional derivative formula $\frac{\partial}{\partial t}\psi(P + t(\overline{P} - P)) \mid_{t=0} = \int \varphi(z; P) d(\overline{P} - P)(z)$. (W.l.o.g. we assume $\varphi(z; P)$ is centered so that $\int \varphi(z; P) dP(z) = 0$.) Here, $-\int \varphi(z; \overline{P}) dP(z)$ is the first-order term, and $R_2(\overline{P}, P)$ is the second-order remainder term, which only depends on products or squares of differences between $P, \overline{P}$.

When ψ is the ATE as in our setting, ψ admits such an expansion [27]

$$\varphi(Z; \overline{P}) := \frac{A}{\overline{\pi}(X)}(Y - \overline{\mu}(X)) + \overline{\mu}(X) - \psi(\overline{P}), \tag{56}$$

where $\overline{\pi}(x) := \overline{P}[A = 1 \mid X]$ and $\overline{\mu}(x) := \overline{P}[Y \mid A = 1, X]$. In particular, we get the following explicit formula for the second-order term

$$R_2(\overline{P}, P) := \int \pi(x) \left(\frac{1}{\overline{\pi}(x)} - \frac{1}{\pi(x)}\right) (\overline{\mu}(x) - \mu(x))\, dP(x). \tag{57}$$

We will apply this to our C-Learner estimator as defined in Section 6. Recall that $\widehat{\mu}_{-k,n}$ is trained to predict outcome given X and $A = 1$ on $P_{-k,n}$, and $\widehat{\pi}^{C'}_{-k,n}$ is trained to predict

treatment A given X using $P_{-k,n}$ under the constraint that $P_{k,n}\left[\left(1-\frac{A}{\widehat{\pi}^{C'}_{-k,n}(X)}\right)\widehat{\mu}_{-k,n}(X)\right]=0$. Our dual C-Learner estimator is the mean of plug-in estimators across folds: for each fold, write $\widehat{\psi}^{C'}_{k,n}=\psi(\widehat{P}^{C'}_{k,n})$ so the C-Learner estimate is the average $\widehat{\psi}^{C'}_{n}=\frac{1}{K}\sum_{k=1}^{K}\widehat{\psi}^{C'}_{k,n}$.

Noting that any distribution decomposes $\overline{P}=\overline{P}_X\times\overline{P}_{A|X}\times\overline{P}_{Y|A,X}$ and $\psi(\overline{P})=\overline{P}_X[\mu(X)]$, the following definitions

$$\begin{aligned}\widehat{P}^{C'}_{X;k,n}&:=P_{X;k,n}\\ \widehat{P}^{C'}_{A|X;k,n}[A=1\mid X=x]&:=\widehat{\pi}^{C'}_{-k,n}(x)\\ \widehat{P}^{C'}_{Y|A,X;k,n}[Y\mid A=1,X=x]&:=\widehat{\mu}_{-k,n}(x)\end{aligned}$$

provide a well-defined joint distribution $\widehat{P}^{C}_{k,n}$.

For each data fold k, we use the distributional Taylor expansion above, where we replace $\overline{P}$ with the joint distribution $\widehat{P}^{C'}_{k,n}$

$$\begin{aligned}\psi(\widehat{P}^{C'}_{k,n})-\psi(P)&=-P\varphi(Z;\widehat{P}^{C'}_{k,n})+R_2(\widehat{P}^{C'}_{k,n},P)\\ &=(P_{k,n}-P)\varphi(Z;P)-P_{k,n}\varphi(Z;\widehat{P}^{C'}_{k,n})\\ &\quad+(P_{k,n}-P)(\varphi(Z;\widehat{P}^{C'}_{k,n})-\varphi(Z;P))+R_2(\widehat{P}^{C'}_{k,n},P).\end{aligned}\tag{58}$$

Observe that by using Equation (56) and the definition of the dual C-Learner,

$$\begin{aligned}P_{k,n}\varphi(Z;\widehat{P}^{C'}_{k,n})&=P_{k,n}\left[\frac{A}{\widehat{\pi}^{C'}_{-k,n}(X)}(Y-\widehat{\mu}_{-k,n}(X))+\widehat{\mu}_{-k,n}(X)-\psi(\widehat{P}^{C'}_{k,n})\right]\\ &=\underbrace{P_{k,n}\left[\left(1-\frac{A}{\widehat{\pi}^{C'}_{-k,n}(X)}\right)\widehat{\mu}_{-k,n}(X)\right]}_{=0\text{ by dual C-Learner constraint}}+\underbrace{P_{k,n}\left[\frac{A}{\widehat{\pi}^{C'}_{-k,n}(X)}Y-\psi(\widehat{P}^{C'}_{k,n})\right]}_{=0\text{ by (54)}}.\end{aligned}$$

Taking the average of Equation (58) over $k=1,\ldots,K$, we can write the error $\widehat{\psi}^{C'}_{n}-\psi$ as the sum of three terms.

$$\widehat{\psi}^{C'}_{n}-\psi=\frac{1}{K}\sum_{k=1}^{K}\underbrace{(P_{k,n}-P)\varphi(Z;P)}_{S^*_k}$$

$$+ \frac{1}{K}\sum_{k=1}^{K} \underbrace{(P_{k,n} - P)\left(\varphi(Z;\widehat{P}_{k,n}^{C'}) - \varphi(Z;P)\right)}_{T_{1k}} + \frac{1}{K}\sum_{k=1}^{K} \underbrace{R_2(\widehat{P}_{k,n}^{C'}, P)}_{T_{2k}}. \tag{59}$$

Using the decomposition (59), we write

$$S^* = \frac{1}{K}\sum_{k=1}^{K} S_k^*, \quad T_1 = \frac{1}{K}\sum_{k=1}^{K} T_{1k}, \quad T_2 = \frac{1}{K}\sum_{k=1}^{K} T_{2k}, \tag{60}$$

so that $\widehat{\psi}_n^C - \psi = S^* + T_1 + T_2$. We address the terms S^*, T_1 and T_2 separately. The first term can be rewritten as

$$S^* = \frac{1}{K}\sum_{i=1}^{K}(P_{k,n} - P)\varphi(Z;P) = (P_n - P)\varphi(Z;P)$$

so that by the central limit theorem,

$$\sqrt{n}S^* \overset{d}{\rightsquigarrow} N\left(0, \mathrm{Var}_P(\varphi(Z;P))\right).$$

Observe that this quantity depends only on ψ and P, so that it cannot be made smaller by choice of estimator. If the variance of an estimator for ψ is $\mathrm{Var}_P(\varphi(Z;P))$, then it is semiparametrically efficient in the local asymptotic minimax sense (Theorem 25.21 of [59]). Thus, it suffices to show that the rest of the terms, T_1 and T_2, are $o_P(n^{-1/2})$, so that

$$\sqrt{n}(\widehat{\psi}_n^C - \psi) = \sqrt{n}S^* + o_P(1) \overset{d}{\rightsquigarrow} N\left(0, \mathrm{Var}_P(\varphi(Z;P))\right).$$

For a fixed k, $|T_{1k}| = o_P(n^{-1/2})$ by Assumption J so that $|T_1| = o_P(n^{-1/2})$ as desired. The second-order remainder term in the distributional Taylor expansion (55) where we replace $\overline{P}$ with $\widehat{P}_{k,n}^{C'}$ is

$$T_{2k} = \int \pi(x)\left(\frac{1}{\widehat{\pi}_{-k,n}^{C'}(x)} - \frac{1}{\pi(x)}\right)\left(\widehat{\mu}_{-k,n}(x) - \mu(x)\right) dP(x).$$

Under the overlap assumption (Assumption A) and Cauchy-Schwarz,

$$\begin{aligned} |T_{2k}| &\leq \frac{1}{\eta} \int \left| \widehat{\pi}^{C'}_{-k,n}(x) - \pi(x) \right| \left| \widehat{\mu}_{-k,n}(x) - \mu(x) \right| \, dP(x) \\ &\leq \frac{1}{\eta} \left\| \widehat{\pi}^{C'}_{-k,n} - \pi \right\|_{L_2(P)} \left\| \widehat{\mu}_{-k,n} - \mu \right\|_{L_2(P)} . \end{aligned} \tag{61}$$

By Assumption I, $|T_{2k}| = o_P(n^{-1/2})$ so that $|T_2| = o_P(n^{-1/2})$ as desired. □

D.2 Double robustness of dual C-Learner

We state assumptions for double robustness for the dual C-Learner, as well as the theorem. Assumption K is the counterpart to Assumption E, and Theorem 6 is the counterpart to Theorem 3.

Assumption K (At least one of $\widehat{\pi}^{C'}, \widehat{\mu}$ is consistent)**.** *For all k, the product of the errors for the outcome and propensity models decays as*

$$\| \widehat{\pi}^{C'}_{-k,n} - \pi \|_{L_2(P)} \cdot \| \widehat{\mu}_{-k,n} - \mu \|_{L_2(P)} = o_P(1).$$

Using Assumption K in place of Assumption B, we arrive at the following result:

Theorem 6 (Dual C-Learner is doubly robust)**.** *The dual C-Learner (53) is consistent under Assumptions A, C, J, and K.*

Showing double robustness of the dual C-Learner is completely analogous to Section B.2.

E Additional Details and Results of Experiments

See Section G for how point estimates and confidence intervals are calculated for the estimators in Section 5.

E.1 Experiments for Kang and Schafer [31]

E.1.1 Implementation Details

Linear Outcome Models. Here, $P_{\text{train}} = P_{\text{eval}}$, as consistent with the original paper by Kang and Schafer [31][7]. There is no P_{val} as there are no hyperparameters to tune.

For the direct method and initial outcome model for AIPW, self-normalized AIPW, and TMLE, we simply regress the dependent variable Y on the observed covariates X using the samples with labels. We use all the samples available to fit a logistic regression model for the treatment variable A (outcome was observed) using the covariates X. Following [31], in both models, we omit the intercept term when fitting the propensity scores and initial outcome model. Our main insights do not change if an intercept is added to the outcome and propensity models.

We run datasets with 200 and 1000 observations, with and without a truncation threshold of 5%. For each configuration, we generate 1000 seeds.

Table 7 displays the results for the bias, mean absolute error, root mean squared error (RMSE), and median absolute error over 1000 simulations with sample sizes equal to 200 and 1000. We also include the results when truncating to 5% is performed, meaning that $\widehat{\pi}(X)$ is truncated to have a lower bound of 0.05. Note that truncation greatly improves the performance of propensity-based methods such as AIPW and TMLE. Without truncation, C-Learner is the best method across all samples when compared to other unbiased methods and is outperformed only by the direct method. When truncation is introduced, the C-Learner is similar in performance to other asymptotically optimal methods.

For the coverage computation, we construct the confidence intervals as described in Section G and empirically check for each method if it covers the true population mean. Table 8 presents the coverage results for a 95% confidence level. It is worth noticing that coverage results greatly deteriorate when the sample size increases for all asymptotically optimal methods, most likely because both the outcome model and the propensity model are mispecified, and consistency is ensured when at least one of the models is consistent (see Section 6).

Finally, Table 10 displays the results for bias and mean absolute error (MAE) for

[7] Although sample splitting could be better, we have chosen to replicate [31] as closely as possible.

estimating $P[Y(0)]$ rather than $P[Y(1)]$. In this case, all the asymptotic optimal methods perform very similarly, and all are better than the direct method with OLS. To conclude, we emphasize that the C-Learner is the only asymptotically optimal method that demonstrated good performance across all configurations of data size, truncation procedure, and data-generating process by either achieving comparable performance to other asymptotically optimal methods, or by improving performance, sometimes by an order of magnitude.

Linear Models with Logistic Link. In order to instanciate the TMLE and the C-Learner using the logistic link, first we normalize the observed outcome Y. In particular, for the observed dataset we compute $Y_{\max} = (1+\alpha)\max_{i:A_i=1}\{Y_i\}$ and $Y_{\min} = (1-\alpha)\min_{i:A_i=1}\{Y_i\}$, where the scaling parameter α accounts for the fact that for the observed dataset, both the empirical maximum and empirical minimum are most likely biased. We chose this procedure to match previous work in the literature, such as the one proposed in Van der Laan and Rose [57], and it does not mean to provide unbiased estimators for such estimates. Observed outcomes are then rescaled by computing $\tilde{Y}_i = (Y_i - Y_{\min})/(Y_{\max} - Y_{\min})$. In the experiments reported in Section 5, we use $\alpha = .1$ as this was the parameter used in Van der Laan and Rose [57] when analyzing the same dataset. Different values of α ranging from 0.01 to 0.2 leads to the same insights we provide in the main text, with the C-Learner-L dominating the MAE obtained with TMLE-L. The default α for the `tmle` package available in R [24] is 0.01.

TMLE-L. Abusing notation, define $\sigma(X) := 1/(1+\exp(-X))$, which is the logistic link function. As an starting point, one solve for the fractional logistic regression using the scaled outcomes $\tilde{Y}$:

$$\min_{\theta}\left\{-\sum_{i=1}^{n} A_i\left[\tilde{Y}_i \log\sigma(\theta^\top x_i) + (1-\tilde{Y}_i)\log(1-\sigma(\theta^\top X_i))\right]\right\},$$

which leads to $\widehat{\mu}(X_i) = 1/(1+\exp(-X^\top\theta))$. The second step is to define a parametric submodel and find the optimal fluctuation parameter ϵ. Define

$$\widehat{\mu}(X_i, H_i, \epsilon) := \frac{1}{1+e^{-\log\frac{\widehat{\mu}(X_i)}{1-\widehat{\mu}(X_i)}+\epsilon H_i}},$$

and note that $\widehat{\mu}(X_i, H_i, \epsilon)$ is a shift of $\widehat{\mu}(X_i)$ under the logit transformation in the direction of H_i. Therefore, under the logistic link, the optimal fluctuation can be fitted by solving

$$\epsilon^\star := \underset{\epsilon \in \mathbb{R}}{\operatorname{argmin}} \quad -\frac{1}{n} \sum_{i=1}^{n} A_i \left(\tilde{Y}_i \log \widehat{\mu}(X_i, H_i, \epsilon) - (1 - \tilde{Y}_i) \log(1 - \widehat{\mu}(X_i, H_i, \epsilon)) \right). \tag{62}$$

It is straightforward to verify that the first-order conditions for ϵ imply that the empirical evaluation of (4) at $\widehat{\mu}(A_i, X_i, H_i, \epsilon^\star)$ is zero. Next, the TMLE procedure is executed as usual by evaluating the fluctuated model on the whole sample. To implement TMLE with the logistic link, we use the `tmle` package available in R [24].

C-Learner-L. For the C-Learner-L implementation we solve

$$\theta^\star = \underset{\theta}{\operatorname{argmin}} \; -\sum_{i=1}^{n} A_i \left[\tilde{Y}_i \log \sigma(\theta^\top X_i) + (1 - \tilde{Y}_i) \log(1 - \sigma(\theta^\top X_i)) \right] \quad \text{such that}$$
$$\sum_{i=1}^{n} \tfrac{A_i}{\widehat{\pi}(X_i)} \left(\tilde{Y}_i - \sigma(\theta^\top X_i) \right) = 0.$$

Note that this is a convex optimization problem with nonlinear constraints. In order to solve it, we use a Sequential Least Squares Programming algorithm available in the R package `nloptr` [43].

C-Learner-Dual And Covariate Balancing Methods We describe the implementation of the C-Learner-Dual for linear propensity models under the logistic link. We also discuss the implementation of the benchmark methods. For all models, we also implement their self-normalized version, in which the propensities sum to one.

CBPS We implement the CBPS approach of [28] using the `CBPS` package available in R [24]. We use a linear specification and the over-identified model that is optimized for both propensity score fitting and the balancing constraint with the two-step procedure, following the same procedure as in the package documentation for the KS dataset.

Linear Fluctuation of Propensity Scores We implement a one-step calibration of the initial propensity scores via a low-dimensional fluctuation. The procedure resembles in spirit the parametric submodel approach of TMLE for outcome models for achieving the semiparametric variance bound. We follow [44]'s construction for $\widehat{\mu}_{LIK2}$: Define the calibration covariates and form the extended propensity model

$$\omega_i(\lambda) = \hat{\pi}_i + H_i^\top \lambda, \quad \ell(\lambda) = \sum_{i=1}^{n} \big[A_i \log \omega_i + (1 - A_i) \log(1 - \omega_i)\big].$$

where $H_i = (1 - \widehat{\pi}(X_i))\widehat{\mu}(X_i)$ and π_i is the vector of propensities fitted by standard MLE. We maximize likelihood $\ell(\lambda)$ with respect to λ to obtain $\hat{\lambda}$. In particular, we use the Nelder-Mead method available in the package `optim` in R.

Next, we fix ω and proceed to the second step, and we solve

$$\max_{\lambda_{step}} \sum_i \left[T \frac{\log(\pi_i + \lambda'_{step}[1, \mu(X_i)]^\top) - \log(\omega_i)}{1 - \pi_i} - \lambda'_{step}[1, \mu(X_i)]^\top \right].$$

Define $\omega_{step} = \pi_i + \lambda'_{step}[1, \mu(X_i)]^\top$. The first order condition implies that

$$\frac{1}{n} \sum_{i=1}^{n} \Big(1 - \frac{A_i}{\omega_{step}(X_i)}\Big) \hat{\mu}(X_i) \;=\; 0,$$

which is the C-Learner-Dual constraint.

Similar to the C-Learner, the resulting estimator attains the semiparametric variance bound, and is doubly robust. Our implementation for linear fluctuations differs from [44]'s construction of $\widehat{\mu}_{LIK2}$ only in that we omit the h_2 component in the first step, which, in their paper, makes the resulting estimator sample-bounded (in the sense that estimator values are within the range of outcome values seen in the sample); this omission is for a more fair comparison between estimators.

C-Learner-Dual First, we parametrize the propensity-score model by

$$\pi_{\beta_{\text{C-Learner-Dual}}}(X_i) \;=\; \frac{\exp\big(X_i^\top \beta_{\text{C-Learner-Dual}}\big)}{1 + \exp\big(X_i^\top \beta_{\text{C-Learner-Dual}}\big)},$$

and fit $\beta_{\text{C-Learner-Dual}}$ by maximizing the Bernoulli log-likelihood on the treatment indicators A_i. Equivalently, we minimize the negative log-likelihood

$$-\ell\big(\beta_{\text{C-Learner-Dual}}\big) = -\sum_{i=1}^{n}\Big[A_i\,\eta_i - \log\big(1+e^{\eta_i}\big)\Big], \qquad \eta_i \;=\; X_i^{\top}\beta_{\text{C-Learner-Dual}}.$$

Because this objective is smooth and convex, we also supply its gradient in closed form:

$$\nabla_{\beta_{\text{C-Learner-Dual}}}\big[-\ell(\beta_{\text{C-Learner-Dual}})\big] = -\,X^{\top}\big(A - p(\beta_{\text{C-Learner-Dual}})\big), \quad p(\beta_{\text{C-Learner-Dual}})_i = \frac{e^{\eta_i}}{1+e^{\eta_i}}.$$

Next, to ensure semiparametric efficiency of the resulting inverse–propensity-weighted estimator, we impose the finite-sample balance constraint

$$\frac{1}{n}\sum_{i=1}^{n}\Big(1 - \frac{A_i}{\pi_{\beta_{\text{C-Learner-Dual}}}(X_i)}\Big)\,\hat{\mu}(X_i) \;=\; 0,$$

where $\hat{\mu}(X_i)$ is a fixed outcome-model prediction. Rearranging shows this is equivalent (up to an additive constant) to

$$\sum_{i=1}^{n} A_i\,\hat{\mu}(X_i)\,e^{-X_i^{\top}\beta_{\text{C-Learner-Dual}}} \;-\; \sum_{i=1}^{n}(1-A_i)\,\hat{\mu}(X_i) \;=\; 0.$$

We likewise provide the Jacobian of this constraint:

$$\nabla_{\beta_{\text{C-Learner-Dual}}}\big[\text{constraint}\big] = -\sum_{i=1}^{n} A_i\,\hat{\mu}(X_i)\,e^{-X_i^{\top}\beta_{\text{C-Learner-Dual}}}\,X_i.$$

Because we now have a convex, differentiable objective with one smooth equality constraint, we solve it using an augmented-Lagrangian scheme. In particular, we use the `nloptr` package ([43]) with the NLOPT_LD_AUGLAG solver. Tolerance and maximum number of iterations are set to 10^{-8} and 1000, respectively. Finally, the solution $\widehat{\beta}_{\text{C-Learner-Dual}}$ defines the dual C-Learner propensity estimate

$$\widehat{\pi}^{C}(X_i) = \frac{\exp\big(X_i^{\top}\widehat{\beta}_{\text{C-Learner-Dual}}\big)}{1+\exp\big(X_i^{\top}\widehat{\beta}_{\text{C-Learner-Dual}}\big)},$$

and the average treatment effect is then estimated by

$$\widehat{\psi} = \frac{1}{n} \sum_{i=1}^{n} \frac{A_i}{\widehat{\pi}^C(X_i)} Y_i.$$

Gradient Boosted Regression Tree Outcome Models. We demonstrate the flexibility and performance of the C-Learner in which we instantiate C-Learner using gradient boosted regression trees using the XGBoost package [12] with a custom objective, as outlined in Section 4.3. $\widehat{\pi}$ is fit as a logistic regression on covariates X using L1 regularization (LASSO).

For each seed, and sample size, we randomly take half of the data for P_{train} and half of the data for P_{eval}. For the first phase of Algorithm 1, we perform hyperparameter tuning using $P_{\text{val}} = P_{\text{eval}}$. Hyperparameter tuning is performed using a grid search for the following parameters: learning rate (0.01, 0.05, 0.1, 0.2), feature subsample by tree (0.5, 0.8, 1), and max tree depth (3, 4, 5). We also set the maximum number of weak learners to 2 thousand, and we perform early stopping using MSE loss on P_{val} for 20 rounds. Hyperparameter tuning is performed separately for C-Learner and the initial outcome model.

For the second phase of Algorithm 1, we use the set of hyperparameters found in the first stage. The weak learners are fitted using P_{eval} and $P_{\text{val}} = P_{\text{eval}}$. In order to avoid overfitting in the targeting step, we use a subsampling of 50% and early stopping after 20 rounds.

In Table 9 we provided additional details with truncation in order to understand the impact in the results of AIPW and TMLE that produced unreliable estimates for sample sizes of 200 and 1000 as well as the impact of truncation in the other estimates. When "small" truncation is performed (0.1%), C-Learner is still better than any other estimator. When truncation to 5% is performed, TMLE achieves the best pointwise performance, and C-Leaner and AIPW-SN are statically equal in terms of MAE.

Finally, in Table 11 we present the coverage results for the estimators computed without truncation and a sample size of 200 and 1000 presented in the main text on Table 2. For a sample size of 200, asymptotically optimal methods have similar coverage and substantially lower than the target of 95%. When the sample size increases, the coverage results of C-Learner deteriorate, although it still presents the best performance in terms of MAE.

E.1.2 Estimator Performance When Varying Overlap Between Treatment and Control

In order to test the sensitivity of different asymptotic optimal methods with respect to the propensity score, we modify the probability of treatment in the data generation by introducing a scaling parameter c so that

$$\pi(\xi) = \frac{\exp(c(-\xi_1 + 0.5\xi_2 - 0.25\xi_3 - 0.1\xi_4))}{1 + \exp(c(-\xi_1 + 0.5\xi_2 - 0.25\xi_3 - 0.1\xi_4))}, \tag{63}$$

where we take $c \in \{0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75\}$. The case in which $c = 1$ matches the standard setup of [31]. As discussed in Section 5.1, the closer the value of the scaling c to 0, the more concentrated the treatment probabilities are to 50%. In the other direction, the higher the value of c, the more extreme propensities are observed, increasing the variance of the propensity scores, pushing the overlap assumption to the limit and making the problem of causal estimation more challenging.

For each value of the parameter c, we generate 1000 seeds of sample size 200 with no truncation. We use linear outcome models and logistic propensity models. The results for the mean absolute error are displayed in Figure 4. In Table 5 and Table 6 we show the main statistics for the fitted propensity scores $\widehat{\pi}(X)$ and $1\widehat{\pi}(X)$ as a function of the scaling parameter. The statistics are taken over all observations (treated and non-treated).

Scaling c	Stdev $\widehat{\pi}(X)$	Min $\widehat{\pi}(X)$	Max $\widehat{\pi}(X)$	CVar $\widehat{\pi}(X)$
0.25	0.08 (0.001)	0.2021 (0.003)	0.70 (0.003)	0.296 (0.003)
0.50	0.13 (0.001)	0.0756 (0.002)	0.77 (0.002)	0.186 (0.002)
0.75	0.17 (0.001)	0.0240 (0.001)	0.83 (0.002)	0.102 (0.002)
1.00	0.21 (0.001)	0.0066 (0.000)	0.88 (0.002)	0.055 (0.001)
1.25	0.24 (0.001)	0.0018 (0.000)	0.92 (0.001)	0.029 (0.001)
1.50	0.26 (0.001)	0.0006 (0.000)	0.94 (0.001)	0.016 (0.001)
1.75	0.28 (0.001)	0.0002 (0.000)	0.96 (0.001)	0.009 (0.000)

Table 5. Summary statistics of $\widehat{\pi}(X)$ for different values of scaling parameter c from Equation (63). We show means across 1000 dataset draws of size 200, with standard error in parentheses. CVar refers to the mean of the 5% of smallest values.

Scaling c	Stdev $1/\widehat{\pi}(X)$	Min $1/\widehat{\pi}(X)$	Max $1/\widehat{\pi}(X)$	CVar $1/\widehat{\pi}(X)$
0.25	0.5 (0)	1.42 (0.005)	5 (1.10)	4 (0.128)
0.50	1.3 (1)	1.30 (0.004)	13 (7.57)	6 (1.05)
0.75	3.9 (11)	1.20 (0.003)	42 (148)	15 (16.3)
1.00	13.2 (1016)	1.13 (0.002)	152 (14325)	41 (1453)
1.25	44.0 (16634)	1.09 (0.001)	543 (235232)	117 (23538)
1.50	136.3 (548595)	1.06 (0.001)	1692 (7758318)	348 (775940)
1.75	440.1 (94410206)	1.04 (0.001)	5585 (1335161973)	1003 (133516504)

Table 6. Summary statistics of $1/\widehat{\pi}(X)$ for different values of scaling parameter c from Equation (63). We show means across 1000 dataset draws of size 200, with standard error in parentheses. CVar refers to the mean of the 5% of largest values.

E.1.3 Additional Results

(a) $N = 200$, no truncation

Method	Bias		Mean Abs Err		RMSE		Median Abs Err
Direct	-0.00	(0.103)	2.60	(0.061)	3.24	(0.457)	2.13
IPW	22.10	(2.579)	27.28	(2.52)	84.45	(2733)	9.87
IPW-SN	3.36	(0.292)	5.42	(0.260)	9.83	(13.45)	3.01
AIPW	-5.08	(0.474)	6.16	(0.461)	15.82	(87.1)	3.43
AIPW-SN	-3.65	(0.203)	4.73	(0.179)	7.38	(6.14)	3.26
TMLE	-111.59	(41.073)	112.15	(41.1)	1302.98	(10^6)	3.95
C-Learner	**-2.45**	(0.120)	**3.57**	(0.088)	**4.52**	(0.912)	**2.93**

(b) $N = 1000$, no truncation

Method	Bias		Mean Abs Err		RMSE		Median Abs Err
Direct	-0.43	(0.044)	1.17	(0.028)	1.46	(0.095)	1.00
IPW	105.46	(59.843)	105.67	(59.8)	1894.40	(10^6)	17.95
IPW-SN	6.83	(0.331)	7.02	(0.327)	12.50	(24.0)	**4.09**
AIPW	-41.37	(24.821)	41.39	(24.8)	785.60	(10^5)	5.22
AIPW-SN	-8.35	(0.431)	8.37	(0.430)	15.97	(46.4)	4.92
TMLE	-17.51	(3.493)	17.51	(3.49)	111.77	(10^4)	4.25
C-Learner	**-4.40**	(0.077)	**4.42**	(0.076)	**5.03**	(0.795)	4.21

(c) $N = 200$, truncation threshold 5%

Method	Bias		Mean Abs Err		RMSE		Median Abs Err
Direct	-0.00	(0.103)	2.60	(0.061)	3.24	(0.457)	2.13
IPW	5.13	(0.398)	10.47	(0.275)	13.60	(10.055)	8.35
IPW-SN	**1.17**	(0.132)	3.33	(0.087)	4.32	(0.969)	**2.59**
AIPW	-2.45	(0.121)	3.59	(0.088)	4.54	(0.913)	3.02
AIPW-SN	-2.37	(0.118)	3.50	(0.085)	4.42	(0.859)	2.92
TMLE	-2.01	(0.107)	**3.15**	(0.075)	**3.95**	(0.665)	2.62
C-Learner	-2.15	(0.112)	3.31	(0.079)	4.14	(0.734)	2.83

(d) $N = 1000$, truncation threshold 5%

Method	Bias		Mean Abs Err		RMSE		Median Abs Err
Direct	-0.43	(0.044)	1.17	(0.028)	1.46	(0.095)	1.00
IPW	8.35	(0.179)	8.67	(0.163)	10.08	(3.413)	8.23
IPW-SN	**1.95**	(0.062)	2.27	(0.050)	2.76	(0.293)	**2.01**
AIPW	-3.41	(0.053)	3.44	(0.051)	3.80	(0.378)	3.44
AIPW-SN	-3.31	(0.052)	3.34	(0.050)	3.70	(0.358)	3.35
TMLE	-2.81	(0.046)	**2.85**	(0.044)	**3.17**	(0.274)	2.86
C-Learner	-3.07	(0.049)	3.10	(0.047)	3.43	(0.310)	3.10

Table 7. Estimator performance in 1000 tabular simulations for the linear specification of outcome models. "truncation" refers to truncation $\widehat{\pi}(X)$ away from 0. Standard error is in parentheses. Asymptotically optimal methods are listed beneath the horizontal divider. We highlight the best-performing overall method *other than the direct method* in **bold**.

	$N = 200$		$N = 1000$	
Method	No truncation	5% truncation	No truncation	5% truncation
Direct	0.89	0.89	0.88	0.88
IPW	1.00	1.00	1.00	1.00
IPW-SN	1.00	1.00	1.00	1.00
AIPW	0.88	0.90	0.59	0.56
AIPW-SN	0.92	0.91	0.72	0.58
TMLE	0.84	0.90	0.48	0.60
C-Learner	0.91	0.90	0.68	0.58

Table 8. Coverage results for 1000 simulations in the linear specification. Confidence intervals were set to achieve 95% confidence. Asymptotically optimal methods are listed beneath the horizontal divider.

(a) $N = 200$, truncation threshold = 5% (b) $N = 200$, truncation threshold = 0.1%

Method	Bias		Mean Abs Err		Bias		Mean Abs Err	
Direct	-6.10	(0.10)	6.18	(0.10)	-6.10	(0.10)	6.18	(0.10)
IPW	4.45	(0.75)	17.8	(0.52)	41	(5.82)	54.2	(5.72)
IPW-SN	-3.40	(0.16)	5.06	(0.10)	-1.10	(0.38)	7.20	(0.31)
Lagrangian	-4.94	(0.10)	5.14	(0.09)	-4.96	(0.10)	5.16	(0.09)
AIPW	**-1.80**	(0.14)	3.85	(0.09)	4.82	(1.27)	10.4	(1.23)
AIPW-SN	-2.10	(0.12)	3.64	(0.08)	**-0.57**	(0.26)	5.02	(0.21)
TMLE	-2.84	(0.10)	**3.51**	(0.08)	-1.42	(0.17)	4.19	(0.12)
C-Learner	-3.10	(0.10)	3.68	(0.08)	-3.24	(0.09)	**3.79**	(0.08)

Table 9. Comparison of estimator performance on misspecified datasets from Kang and Schafer [31] in 1000 tabular simulations using gradient boosted regression trees with 200 samples, 5%, and 0.1% truncation threshold. Asymptotically optimal methods are listed beneath the horizontal divider. We highlight the best-performing method in **bold**. Standard errors are displayed within parentheses to the right of the point estimate. "Lagrangian" refers to only performing the first stage in Algorithm 1.

	(a) $N = 200$				(b) $N = 1000$			
Method	Bias		Mean Abs Err		Bias		Mean Abs Err	
Direct	4.82	(0.091)	4.94	(0.085)	4.80	(0.041)	4.80	(0.041)
IPW	**-0.70**	(0.141)	3.44	(0.092)	**-0.83**	(0.054)	**1.52**	(0.036)
IPW-SN	2.48	(0.097)	**3.23**	(0.072)	2.43	(0.042)	2.47	(0.040)
AIPW	3.23	(0.093)	3.63	(0.077)	3.14	(0.041)	3.15	(0.040)
AIPW-SN	3.22	(0.092)	3.61	(0.076)	3.12	(0.041)	3.13	(0.040)
TMLE	3.11	(0.091)	3.50	(0.076)	3.03	(0.041)	3.04	(0.040)
C-Learner	3.19	(0.092)	3.58	(0.076)	3.09	(0.041)	3.10	(0.040)

Table 10. Results of 1000 simulations for estimating $P[Y(0)]$ instead of $P[Y(1)]$ with the linear specification. The standard error is in parentheses. Asymptotically optimal methods are listed beneath the horizontal divider.

Method	$N = 200$		$N = 1000$	
Direct	0.35	(0.01)	0.12	(0.01)
IPW	1.00	(0.00)	0.98	(0.01)
IPW-SN	0.85	(0.01)	1.00	(0.00)
Lagrangian	0.84	(0.01)	0.71	(0.01)
AIPW	0.85	(0.01)	0.85	(0.01)
AIPW-SN	0.85	(0.01)	0.85	(0.01)
TMLE	0.86	(0.01)	0.87	(0.01)
C-Learner	0.82	(0.01)	0.77	(0.01)

Table 11. Coverage results for 1000 simulations using gradient boosted regression trees. Confidence intervals were set to achieve 95% confidence. Asymptotically optimal methods are listed beneath the horizontal divider. "Lagrangian" refers to only performing the first stage in Algorithm 1.

Balancing results, linear outcome model As discussed in Section 2, covariate balancing techniques imply constraints, possibly via customized loss functions, to equate the covariates of samples in treatment and control for a given propensity score model $\hat{\pi}$, thus, informing how to constraint or estimate the propensity model $\hat{\pi}$. Naturally, methods that make use of propensity score such as IPW (the basic CBPS approach), AIPW, TMLE and C-Learner, may also benefit from a tailored propensity score function $\hat{\pi}$.

Therefore, we also provide in Table 12 a comparison with the Covariate Balancing Propensity Score (CBPS). We discuss the implementation details of CBPS in Section E.1.1. Roughly speaking, all asymptotically optimal methods improve or maintain its performance when using propensity scores via covariate balancing, despite an increase in the MAE

	(a) $N = 200$				(b) $N = 1000$			
Method	Bias		Mean Abs Err		Bias		Mean Abs Err	
CBPS-IPW	-6.92	(0.26)	8.52	(0.21)	1.76	(0.15)	4.00	(0.10)
CBPS-IPW-SN	-2.98	(0.10)	3.66	(0.08)	-1.55	(0.06)	1.88	(0.04)
CBPS-AIPW	-1.68	(0.36)	3.11	(0.24)	-3.54	(0.37)	3.58	(0.37)
CBPS-AIPW-SN	-1.69	(0.36)	3.11	(0.24)	-3.44	(0.33)	3.49	(0.33)
CBPS-TMLE	-7.20	(5.14)	9.50	(5.10)	-7.54	(2.48)	7.55	(2.48)
CBPS-C-Learner	-1.70	(0.36)	3.10	(0.24)	-3.10	(0.20)	3.15	(0.19)
CBPS-TMLE-L	-2.01	(0.33)	3.07	(0.24)	-2.76	(0.15)	2.79	(0.14)
CBPS-C-Learner-L	-1.85	(0.34)	2.97	(0.24)	-2.76	(0.18)	2.83	(0.17)

Table 12. Comparison of estimator performance on misspecified datasets from Kang and Schafer [31] in 1000 tabular simulations, for linear outcome models (Section 5.1.1) using propensity scores fitted via covariate balancing. We include only methods that makes use of propensity scores. We highlight the best-performing method in **bold**. Standard errors are displayed within parentheses to the right of the point estimate.

standard error. For the sample size of 200, C-Learner with covariate balancing and logistic loss is the best performance method for the MAE point estimate, although most of the asymptotically optimal methods are statistically the same. When the sample size is 1000, covariate balancing with self-normalization is the best performing method.

E.2 CivilComments Experiments (Section 5.2)

E.2.1 Implementation Details

Model and Training Objectives Both propensity and outcome models are a linear layer on top of a DistilBERT featurizer, with either softmax and cross-entropy loss for propensity score, or mean squared error for outcome models. For training outcome models, we only consider training data on which $A = 1$, as only those terms contribute to the loss.

Hyperparameters and Settings For propensity models, we tried learning rates of $\{10^{-3}, 10^{-4}, 10^{-5}\}$ for the setting of $l = 10^{-4}$ and found that a learning rate of 10^{-4} performed the best in terms of val loss. Because of computational constraints, we also used this learning rate for $l = 10^{-2}, 10^{-3}$.

For outcome models (including C-Learner) in the settings of $l \in \{10^{-2}, 10^{-3}, 10^{-4}\}$, we tried learning rates of $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. For C-Learner regularization, we tried

λ taking on values of $\lambda_0 / P_{\text{eval}}[A/\hat{\pi}(X)]^2$ where $\lambda_0 \in \{0, 1, 4, 16, 64, 256\}$. Essentially, the penalty is λ_0 times the square of the bias shift that would be required to satisfy the condition.

Because of computational constraints, for choosing these hyperparameters (learning rate, λ), we select a subset of dataset draws, and choose the best hyperparameters based on each model's criteria on that small set of dataset draws. Then, after the hyperparameters are chosen, we run over the entire set of dataset draws.

While hyperparameters for the (usual) outcome model are chosen to minimize val loss, in contrast, the hyperparameters (learning rate, regularization λ) for the C-Learner outcome model were selected to minimize the *magnitude of the constant shift* at the end of the first epoch, as in Section 4.4. The idea is that if the size of the constant shift is small, then the regularizer is doing a good job of enforcing the constraint. Although one might think that larger regularization λ's would automatically lead to smaller constant shifts, we do not find this to be the case; the best λ in our settings (64) is not the largest value that we try (up to 256). Furthermore, the set of hyperparameter values we choose between in this process are ones that seem to result in reasonable performance, e.g. in terms of MSE on treated units in the validation set.

The best hyperparameters chosen are in Table 15. We trained all models with batch size 64. Learning rates decayed linearly over 10 epochs. Optimization was done using the AdamW optimizer. Weight decay was fixed at 0.1.

We choose the training epoch with the best val loss for each model (cross entropy for propensity, and MSE for outcome). For C-Learner, we choose the epoch with the best val MSE.

E.2.2 Additional Results

We consider three data generating processes as in Section 5.2, where treatment assignment is parameterized by $l = 10^{-4}, 10^{-3}, 10^{-2}$, respectively. Summary statistics of $\widehat{\pi}(X)$ are in Table 13, and summary statistics of $1/\widehat{\pi}(X)$ are in table 14. Hyperparameters for outcome models are in Table 15. Comparisons of estimators in these settings are in Tables 16, 17, 18.

Value of l	Min $\widehat{\pi}(X)$	Max $\widehat{\pi}(X)$	Std $\widehat{\pi}(X)$	CVar $\widehat{\pi}(X)$
$l = 10^{-2}$	0.0014 (0.0001)	0.94 (0.01)	0.15 (0.002)	0.0016 (0.0001)
$l = 10^{-3}$	0.0004 (0.0000)	0.95 (0.01)	0.15 (0.002)	0.0004 (0.0000)
$l = 10^{-4}$	0.0003 (0.0000)	0.96 (0.00)	0.16 (0.002)	0.0003 (0.0000)

Table 13. Summary statistics of $\widehat{\pi}(X)$ for values of hyperparameter $l = 10^{-4}, 10^{-3}, 10^{-2}$ for Section 5.2 experiments. We show means across 100 dataset draws for each value of l, with standard error in parentheses. CVar refers to the mean of the 5% of smallest values.

Value of l	Min $1/\widehat{\pi}(X)$	Max $1/\widehat{\pi}(X)$	Std $1/\widehat{\pi}(X)$	CVar $1/\widehat{\pi}(X)$
$l = 10^{-2}$	1.07 (0.01)	1124.7 (91.0)	237.1 (23.4)	1050.1 (85.3)
$l = 10^{-3}$	1.06 (0.01)	3244.0 (190.5)	757.7 (54.1)	3090.9 (183.5)
$l = 10^{-4}$	1.05 (0.00)	4126.9 (214.6)	980.6 (61.4)	3960.2 (205.7)

Table 14. Summary statistics of $1/\widehat{\pi}(X)$ for values of hyperparameter $l = 10^{-4}, 10^{-3}, 10^{-2}$ for Section 5.2 experiments. We show means across 100 dataset draws for each value of l, with standard error in parentheses. CVar refers to the mean of the 5% of largest values.

	$l = 10^{-4}$	$l = 10^{-3}$	$l = 10^{-2}$
Outcome model learning rate	10^{-3}	10^{-3}	10^{-4}
C-Learner learning rate (best val MSE)	10^{-4}	10^{-4}	10^{-4}
C-Learner learning rate (min bias shift)	10^{-5}	10^{-5}	10^{-5}
C-Learner λ (best val MSE)	0	16	16
C-Learner λ (min bias shift)	64	64	64

Table 15: Hyperparameters for $l = 10^{-4}, 10^{-3}, 10^{-2}$ for Section 5.2

Method	Bias		Mean Abs Err		Coverage	
Direct	0.173	(0.008)	0.177	(0.007)	0.010	(0.001)
IPW	0.504	(0.084)	0.546	(0.081)	0.760	(0.018)
IPW-SN	0.114	(0.017)	0.153	(0.014)	0.890	(0.010)
AIPW	0.084	(0.043)	0.307	(0.032)	0.830	(0.014)
AIPW-SN	0.116	(0.018)	0.161	(0.014)	0.850	(0.013)
TMLE	-1.264	(1.361)	1.802	(1.355)	0.810	(0.015)
C-Learner (best val MSE)	0.103	(0.015)	0.141	(0.011)	0.870	(0.011)
C-Learner (min bias shift)	**0.075**	(0.012)	**0.115**	(0.008)	0.900	(0.009)

Table 16. Comparison of estimators in the CivilComments [16] semi-synthetic dataset over 100 re-drawn datasets, with $l = 10^{-4}$. Confidence intervals were set to achieve 95% confidence. Asymptotically optimal methods are listed beneath the horizontal divider. We highlight the best-performing overall method in **bold**. Standard errors are displayed within parentheses to the right of the point estimate.

Method	Bias		Mean Abs Err		Coverage	
Direct	0.147	(0.028)	0.200	(0.025)	0.000	(0.000)
IPW	0.417	(0.064)	0.437	(0.063)	0.840	(0.013)
IPW-SN	0.079	(0.015)	0.120	(0.013)	0.910	(0.008)
AIPW	**-0.056**	(0.052)	0.344	(0.039)	0.950	(0.005)
AIPW-SN	0.089	(0.020)	0.134	(0.018)	0.950	(0.005)
TMLE	0.074	(0.025)	0.144	(0.022)	0.940	(0.006)
C-Learner (best val MSE)	0.067	(0.012)	0.110	(0.009)	0.980	(0.002)
C-Learner (min bias shift)	0.060	(0.011)	**0.099**	(0.008)	0.990	(0.001)

Table 17. Comparison of estimators in the CivilComments [16] semi-synthetic dataset over 100 re-drawn datasets, with $l = 10^{-3}$. Confidence intervals were set to achieve 95% confidence. Asymptotically optimal methods are listed beneath the horizontal divider. We highlight the best-performing overall method in **bold**. Standard errors are displayed within parentheses to the right of the point estimate.

Method	Bias		Mean Abs Err		Coverage	
Direct	0.091	(0.007)	0.099	(0.006)	0.040	(0.004)
IPW	0.346	(0.041)	0.353	(0.040)	0.710	(0.021)
IPW-SN	0.004	(0.005)	0.038	(0.003)	0.950	(0.005)
AIPW	-0.172	(0.029)	0.220	(0.026)	0.890	(0.010)
AIPW-SN	0.002	(0.005)	**0.040**	(0.003)	1.000	(0.000)
TMLE	0.027	(0.005)	0.042	(0.003)	0.990	(0.001)
C-Learner (best val MSE)	0.009	(0.007)	0.053	(0.004)	1.000	(0.000)
C-Learner (min bias shift)	**0.001**	(0.006)	0.045	(0.003)	1.000	(0.000)

Table 18. Comparison of estimators in the CivilComments [16] semi-synthetic dataset over 100 re-drawn datasets, with $l = 10^{-2}$. Confidence intervals were set to achieve 95% confidence. Asymptotically optimal methods are listed beneath the horizontal divider. We highlight the best-performing *asymptotically optimal* method in **bold**. Standard errors are displayed within parentheses to the right of the point estimate. C-Learner's performance is within standard error of AIPW-SN and TMLE.

C-Learner errors are comparatively more stable for $l = 10^{-4}, 10^{-3}$. For $l = 10^{-2}$, C-Learner is comparable to other asymptotically optimal methods.

E.3 IHDP Tabular Dataset

The Infant Health and Development Program (IHDP) dataset is a tabular semisynthetic dataset [10] that was first introduced as a benchmark for ATE estimation by Hill [26]. IHDP is based on a randomized experiment studying the effect of specialized healthcare

interventions on the cognitive scores of premature infants with low birth weight. The dataset consists of a continuous outcome variable Y, a binary treatment variable A, and 25 covariates X that affect the outcome variable and are also correlated with the treatment assignment. We use 1000 datasets as in [42, 15].

E.3.1 Gradient Boosted Regression Trees

We instantiate C-Learner using gradient boosted regression trees using the XGBoost package [12] with a custom objective, as outlined in Section 4.3. $\widehat{\pi}$ is fit as a logistic regression on covariates X. Different to the applications presented in the main text, we are interested in estimating the ATE $P[Y(1) - Y(0)]$ where $Y(0)$ is nonzero. We discuss two alternatives to instantiate the C-Learner. First, we find an outcome model that takes as input both covariates and the treatment variable, which leads to the following formulation

$$\widehat{\mu}^C_{-k,n} = \operatorname*{argmin}_{\tilde{\mu}\in\mathcal{F}} \left\{ P_{-k,n}[(Y - \tilde{\mu}(A,X))^2] : P_{k,n}\left[\left(\frac{A}{\widehat{\pi}(X)} - \frac{1-A}{1-\widehat{\pi}(X)}\right)(Y - \tilde{\mu}(A,X))\right] = 0 \right\},$$

and the final estimator becomes

$$\widehat{\psi}^C_n := \frac{1}{K}\sum_{k=1}^{K} P_{k,n}[\widehat{\mu}^C_{-k,n}(1,X) - \widehat{\mu}^C_{-k,n}(0,X)].$$

This approach is often referred to as the "S-Learner".

Another alternative is to model the ATE as the difference of two missing outcomes and fit two outcome models for treated vs. non-treated units. This approach is commonly referred to as the "T-Learner". In this case, we estimate two models by solving

$$\widehat{\mu}^C_{-k,n,1} \in \operatorname*{argmin}_{\tilde{\mu}\in\mathcal{F}} \left\{ P_{-k,n}[A(Y - \tilde{\mu}(X))^2] : P_{k,n}\left[\frac{A}{\widehat{\pi}_{-k,n}(X)}(Y - \tilde{\mu}(X))\right] = 0 \right\},$$
$$\widehat{\mu}^C_{-k,n,0} \in \operatorname*{argmin}_{\tilde{\mu}\in\mathcal{F}} \left\{ P_{-k,n}[(1-A)(Y - \tilde{\mu}(X))^2] : P_{k,n}\left[\frac{1-A}{1-\widehat{\pi}_{-k,n}(X)}(Y - \tilde{\mu}(X))\right] = 0 \right\},$$

and the final estimator becomes

$$\widehat{\psi}^C_n := P_{\text{eval}}\frac{1}{K}\sum_{k=1}^{K} P_{k,n}[\widehat{\mu}^C_{-k,n,1}(X) - \widehat{\mu}^C_{-k,n,0}(X)].$$

We empirically observe that the latter approach (T-Learner) is critical for performance in this setting: using a simple XGBoost T-Learner model significantly improves upon sophisticated S-Learner variants, including state-of-the-art methods such as RieszNet, RieszForest, and DragonNet [15, 42].

For consistency with prior work [52, 15, 42], for each dataset run, we split the data in 80% training and 20% for hyperparameter tuning, and use the whole dataset when evaluating the first-order error term, i.e., $P_{\text{eval}} = P_{\text{train}} \cup P_{\text{val}}$. For each outcome model, $\widehat{\mu}^{C}_{-k,n,1}$ and $\widehat{\mu}^{C}_{-k,n,0}$, we follow the same setup described in the Kang & Schafer experiment. Hyperparameter tuning is performed using a grid search for the following parameters: learning rate (0.01, 0.05, 0.1, 0.2), feature subsample by tree (0.5, 0.8, 1), and max tree depth (3, 4, 5). We also set the maximum number of weak learners to 2000, and we perform early stopping using MSE loss on P_{val} for 20 rounds. Hyperparameter tuning is performed separately for C-Learner and the initial outcome model. For the second phase of Algorithm 1, we use the set of hyperparameters found in the first stage. The weak learners are fitted using P_{eval}. In order to avoid overfitting in the targeting step, we use a subsampling of 50% and early stopping after 20 rounds.

The results for the mean absolute error and their respective standard errors are displayed in Table 19. The C-Learner improves upon the direct method while using an identical model class. The asymptotically optimal estimators perform similarly, and C-Learner's advantage over other asymptotically optimal methods is not statistically significant. This is perhaps not surprising in this setting as the propensity weights do not vary as much here, in contrast to Kang & Shafer's example where existing asymptotically optimal methods performed poorly as estimated propensity weights varied dramatically.

E.3.2 Neural Networks

We also instantiate C-Learner as neural networks (Section 4.4) on the IHDP dataset. In particular, we demonstrate how C-Learner can use Riesz representers (A), or equivalently, propensity models in the ATE setting, that are learned by other methods. This demonstrates the versatility of C-Learner as it is able to leverage new methods for learning Riesz representers. In Table 20, we demonstrate C-Learner using Riesz representers learned by RieszNet [15]. C-Learner achieves very similar performance to RieszNet while using the

Method	Bias		Mean Abs Err	
Direct (Boosting)	-0.16	(0.004)	0.188	(0.004)
IPW	-1.50	(0.040)	1.500	(0.040)
IPW-SN	-0.01	(0.003)	0.110	(0.003)
Lagrangian	-0.09	(0.004)	0.144	(0.004)
AIPW	-0.05	(0.002)	0.103	(0.003)
AIPW-SN	-0.04	(0.002)	**0.103**	(0.003)
TMLE	0.007	(0.003)	0.103	(0.003)
C-Learner	**-0.004**	(0.003)	0.104	(0.003)

Table 19. Comparison of gradient boosted regression tree-based estimators in the IHDP semi-synthetic dataset over 1000 simulations. Asymptotically optimal methods are listed beneath the horizontal divider. We highlight the best-performing method in **bold**. Standard errors are displayed within parentheses to the right of the point estimate. "Lagrangian" refers to only performing the first stage in Algorithm 1. C-Learner's performance is within standard error of AIPW-SN.

same Riesz representer. All methods use an outcome model trained in the usual way, except for RieszNet and C-Learner. In all settings where applicable, we use the Riesz representer $(A/\pi(X) - (1-A)/(1-\pi(X))$ learned by RieszNet.

We emphasize that this is not meant to be a performance comparison between RieszNet and C-Learner, for two reasons: first, we present C-Learner as a debiasing framework that is compatible with other methods for causal inference, including RieszNet; and second, we found that propensity scores fitted to the IHDP dataset are not extreme, so we expect C-Learner to perform as well as existing methods, but not to outperform existing methods.

Method	Bias		Mean Abs Err		Coverage	
Direct	-0.005	(-0.000)	0.118	(0.003)	0.783	(0.025)
IPW	-0.789	(-0.025)	0.903	(0.034)	0.455	(0.014)
IPW-SN	-0.449	(-0.014)	0.654	(0.035)	0.819	(0.026)
AIPW	-0.044	(-0.001)	0.106	(0.003)	0.940	(0.030)
AIPW-SN	-0.042	(-0.001)	0.106	(0.003)	0.961	(0.030)
RieszNet ("DR")	**-0.033**	(-0.001)	**0.098**	(0.003)	0.972	(0.031)
C-Learner	-0.038	(-0.001)	0.098	(0.002)	0.955	(0.030)

Table 20. Comparison of neural network based estimators in the IHDP semi-synthetic dataset over 1000 simulations. Asymptotically optimal methods are listed beneath the horizontal divider. We highlight the best-performing *asymptotically optimal* method in **bold**. Standard errors are displayed within parentheses to the right of the point estimate. All methods use an outcome model trained in the usual way, except for RieszNet and C-Learner. In all settings where applicable, we use the Riesz representer learned by RieszNet.

Training Procedure We use lightly modified RieszNet code [15] to generate the Riesz representers used in all methods. We use the same neural network architecture for RieszNet outcome models for C-Learner. We use $P_{\text{train}}, P_{\text{val}}, P_{\text{eval}}$ as in Section E.3.1. Hyperparameters (learning rate, λ) are selected for best mean squared error on P_{val} for each individual dataset. For the training objective, we largely follow Section 4.4, with the following modifications for the full ATE rather than assuming that $Y(0) = 0$: we use the mean squared error across the full dataset (as it was originally for RieszNet), and we have separate penalties and bias shifts for $A = 1$ and $A = 0$. As in RieszNet training, there is a phase of "pre-training" with a higher learning rate, and then a phase with a lower learning rate; for "pre-training" we sweep over learning rates of $\{10^{-3}, 10^{-4}\}$ for 100 epochs; regular training uses a learning rate of 10^{-5} for 600 epochs. Both use the Adam optimizer. For λ we sweep over values of $\{0, 0.01, 0.02, 0.04, 0.1, 0.2, 0.4, 1, 2, 4, 8\}$. Our implementation of RieszNet as a baseline is almost identical to that of the original paper, except for how we set the random seed, for initializing neural networks and for choosing train and test split.

Comparisons with results in RieszNet paper Surprisingly, our mean absolute error results in Section E.3.2, including the ones using regular RieszNet (including their algorithm code and hyperparameters), often outperform the results reported in the original RieszNet paper [15]. One difference between our code and the RieszNet code is how random seeds were set. The random seed affects the train/test split, and also weight initialization for the neural network. We re-ran the original RieszNet code, and then again with a fixed seed (set to 123) for every dataset, and we found that the results were noticeably different. In particular, the RieszNet mean absolute error is smaller with seeds set the second way. Compare Table 21 and Table 22. In our experiments on this setting, we fix seeds in the second way.

Method	Mean absolute error (s.e.)
Doubly robust	0.115 (0.003)
Direct	0.124 (0.004)
IPW	0.801 (0.040)

Table 21: Original RieszNet IHDP results

Method	Mean absolute error (s.e.)
Doubly robust	0.098 (0.003)
Direct	0.118 (0.003)
IPW	0.903 (0.034)

Table 22: RieszNet IHDP results, with fixed seeds

Relative Performance We've seen in Section 5.2 that C-Learner tends to outperform one-step estimation and targeting in settings with low overlap. Here, it merely performs about the same as one-step estimation and targeting. It seems as though overlap is not very low in IHDP, however, based on propensity scores from trained models not taking on extreme values. Using RieszNet's learned Riesz representers, the smallest value for $\min(\widehat{\pi}, 1-\widehat{\pi})$ across all IHDP datasets is 0.11, which is a fair bit of overlap, especially in comparison to, say, the settings with low overlap in Section 5.2 where C-Learner outperformed one-step estimation and targeting methods. Thus, here it is expected that C-Learner matches the performance of other methods. We further emphasize that C-Learner is compatible with methods such as RieszNet, making C-Learner generally applicable. To recap, over all of our experiments, C-Learner appears to perform either comparably to or better than one-step estimation and targeting, and outperform one-step estimation and targeting in settings with less overlap.

F TMLE extensions and additional connections

This section discusses existing TMLE extensions that can mitigate practical instabilities under limited overlap, and additional connections between C-Learner and the TMLE literature. Our paper's main focus is not on the extensions and variations of TMLE, but simply on a new way to debias estimators via constrained optimization, that could potentially be combined with additional assumptions, extensions, variations, etc. Nevertheless, these connections to TMLE and variations of TMLE merit mentioning.

A key benefit of C-Learner is that it reduces instability, for instance, when propensity score estimates become extreme. We compared C-Learner to *basic* forms of one-step estimation and TMLE that do not incorporate additional bounding assumptions or regu-

larization heuristics. The TMLE framework includes a number of practical heuristics and assumptions for dealing with limited overlap and bounded outcomes:

- Alternative loss functions and logistic fluctuations: Standard TMLE implementations in software such as `tmle` [24] by default enforce certain bounds for continuous outcomes via a logistic fluctuation, but can also perform the linear model as well. If outcomes are truly bounded, modeling outcomes using a logistic link will constraint the resulting estimates to be within these bounds; in contrast, methods such as IPW and AIPW do not obey such constraints. TMLE with logistic fluctuations is shown to produce stable estimators [57].

- Regularization of TMLE: regularized variations and extensions of TMLE such as Collaborative TMLE (C-TMLE) [23, 29, 50, 53] have been shown to perform well in challenging settings, including those with low overlap [30, 4]. Collaborative TMLE is a variant of TMLE that selects from a sequence of propensity score models of increasing complexity, based on their ability to improve the loss of the fluctuated outcome model, which can discourage selecting propensity score models with extreme values that hurt estimator stability.

There is a broad literature of other regularized TMLE variants, many of which are designed to construct stable estimators that perform well in finite samples, such as Adaptive TMLE (A-TMLE) [47, 56], which adaptively selects between propensity scores of varying complexity. Additional advancements include TMLE methods based on the highly adaptive lasso (HAL) [6, 7, 48, 58], which offer flexible and robust estimation including empirically in settings with low overlap. Variations of C-Learner can also be complementary to such procedures.

Lastly, we note that the theoretical justifications for asymptotic properties for both C-Learner and TMLE build on the same underlying mathematical frameworks, as both remove the first-order error term in the distributional Taylor expansion, and then show that the second-order remainder term is negligible under suitable assumptions. Proofs of this nature have appeared in the TMLE literature [52, 55, 51, 48, 49]. The main difference between C-Learner and TMLE is how the first-order error term is fixed to zero: in C-Learner we impose this constraint directly within our chosen function class of nuisance

parameters (via constrained optimization), rather than using a specific single-dimensional parametric fluctuation along the least favorable submodel [9] that produces TMLE.

G Point Estimates and Confidence Intervals

We define point estimates and calculate variance for the estimators considered in the experiments. These are used to calculate estimator error, confidence intervals, and coverage in the experiments. First, we combine folds: for all X_i in P_n, if X_i is in the kth fold, then let $\widehat{\pi}_n(X_i) := \widehat{\pi}_{-k,n}(X_i)$. We do the same for $\widehat{\mu}, \widehat{\mu}^C$. This way, we construct $\widehat{\pi}_n, \widehat{\mu}_n, \widehat{\mu}_n^C$ over all of the available data, P_n, and where nuisance parameters are only evaluated on data separate from training.

G.1 Direct Method

The direct method estimator (naive plug-in of the outcome model) is given by

$$\widehat{\psi}_n^{\text{direct}} = \frac{1}{K}\sum_{k=1}^{K} \widehat{\psi}_{k,n}^{\text{direct}} = \frac{1}{K}\sum_{k=1}^{K} P_{k,n}[\widehat{\mu}_{-k,n}(X)] = \frac{1}{n}\sum_{i=1}^{n} \widehat{\mu}_n(X_i). \tag{64}$$

Let Var_n denote the empirical variance, calculated on the sample P_n. Then

$$\text{Var}(\widehat{\psi}_n^{\text{direct}}) = \frac{1}{n}\text{Var}_P(\widehat{\mu}_n(X)) \asymp \frac{1}{n}\text{Var}_n(\widehat{\mu}_n(X)),$$

which we use to estimate the variance of the direct method estimator.

G.2 Inverse Propensity Weighting (IPW)

Under similar notation, the inverse probability weighting method gives the estimator

$$\widehat{\psi}_n^{\text{IPW}} = \frac{1}{K}\sum_{k=1}^{K} \widehat{\psi}_n^{\text{IPW}} = \frac{1}{K}\sum_{k=1}^{K} P_{k,n}\left[\frac{AY}{\widehat{\pi}_{-k,n}(X)}\right] = \frac{1}{n}\sum_{i=1}^{n} \frac{A_i Y_i}{\widehat{\pi}_n(X_i)}$$

The variance of the estimator is given by

$$\mathrm{Var}_P(\widehat{\psi}_n^{\mathrm{IPW}}) = \frac{1}{n}\mathrm{Var}_P\left(\frac{AY}{\widehat{\pi}_n(X)}\right) \asymp \frac{1}{n}\mathrm{Var}_n\left(\frac{AY}{\widehat{\pi}_n(X)}\right).$$

G.3 Self-Normalized IPW

Consider $\widehat{\pi}$. Then the self-normalized IPW estimator (a.k.a. Hajek estimator) is

$$\widehat{\psi}_{\text{IPW-SN}} = \frac{\frac{1}{n}\sum_{i=1}^n \frac{A_i Y_i}{\widehat{\pi}_n(X_i)}}{\frac{1}{n}\sum_{i=1}^n \frac{A_i}{\widehat{\pi}_n(X_i)}}.$$

Let $\widetilde{\pi}_n(X) = \widehat{\pi}_n(X) \cdot \frac{1}{n}\sum \frac{A_i}{\widehat{\pi}_n(X_i)}$ so that

$$\widehat{\psi}_{\text{IPW-SN}} = \frac{1}{n}\sum_{i=1}^n \frac{A_i Y_i}{\widetilde{\pi}_n(X_i)}.$$

Similar to before, the variance of the estimator is given by

$$\mathrm{Var}_P(\widehat{\psi}_{k,n}^{\text{IPW-SN}}) = \frac{1}{n}\mathrm{Var}_P\left(\frac{AY}{\widetilde{\pi}_n(X)}\right) \asymp \frac{1}{n}\mathrm{Var}_n\left(\frac{AY}{\widetilde{\pi}_n(X)}\right).$$

G.4 Any De-Biased Method

This applies to the AIPW, self-normalized AIPW, TMLE and C-Learner. These are first-order corrected (de-biased) and have asymptotic variance as described in Theorem 2. We thus calculate the empirical variance of the plug-in in Theorem 2.