**ECON 7377 Microeconometrics** **Spring 2026**

# Heterogeneous Treatment Effects

## Contents



**Notation.** We work in the potential outcomes framework introduced in the lecture on causal inference. We observe i.i.d. data $\{(Y_i, D_i, X_i)\}_{i=1}^n$, where $Y_i \in \mathbb{R}$ is the outcome, $D_i \in \{0, 1\}$ is the binary treatment indicator, and $X_i \in \mathcal{X} \subseteq \mathbb{R}^p$ is a vector of pre-treatment covariates. The potential outcomes are $Y_i(1)$ and $Y_i(0)$, with observed outcome $Y_i = D_i Y_i(1) + (1 - D_i)Y_i(0)$. We maintain the standard assumptions:

- *SUTVA*: no interference between units.
- *Unconfoundedness*: $(Y_i(1), Y_i(0)) \perp\!\!\!\perp D_i \mid X_i$.
- *Overlap*: $0 < p(x) < 1$ for all $x \in \mathcal{X}$, where $p(x) = \Pr[D_i = 1 \mid X_i = x]$ is the propensity score.

We write $\mu_d(x) = \mathbb{E}[Y_i \mid D_i = d, X_i = x]$ for the conditional response surfaces and $m(x) = \mathbb{E}[Y_i \mid X_i = x]$ for the marginal conditional expectation of the outcome, integrating over the treatment.

# 1 Conditional Average Treatment Effects

## 1.1 From ATE to CATE

- In the lecture on causal inference, we focused on the *average treatment effect* $\text{ATE} = \mathbb{E}[Y_i(1) - Y_i(0)]$, a single summary of how a treatment affects the population on average. In many applications, however, we want to go beyond the average and understand *who benefits most* (or least) from the treatment.

- Examples:

  - In personalized medicine, we may want to identify groups of patients more likely to benefit from a drug.
  - In marketing, we may want to target customers most likely to respond to an offer.
  - In public policy, a program administrator may want to allocate a scarce intervention to units where it has the largest impact.

- At first glance, one might target the individual treatment effect (ITE) $\Delta_i = Y_i(1) - Y_i(0)$. However, since we can never observe both potential outcomes for the same unit, the ITE is generally not point-identified, even under unconfoundedness.

A more practical way to quantify treatment heterogeneity is via the following object.

**Definition 1** (Conditional Average Treatment Effect)**.** The *conditional average treatment effect (CATE)* is the function

$$\tau(x) = \mathbb{E}\left[Y_i(1) - Y_i(0) \mid X_i = x\right]. \tag{1}$$

- The CATE is still an average effect, but it now varies with covariates $X_i$. It is point-identified under unconfoundedness:

  $$\tau(x) = \mu_1(x) - \mu_0(x), \tag{2}$$

  where $\mu_d(x) = \mathbb{E}[Y_i \mid D_i = d, X_i = x]$ are the conditional response surfaces.

- The ATE aggregates over the covariate distribution: $\text{ATE} = \mathbb{E}[\tau(X_i)]$. The CATE $\tau(\cdot)$ is a *function*, whereas ATE is a scalar. The definition of the CATE depends on which covariates $X_i$ are used: conditioning on a richer set of covariates produces a more expressive CATE function that captures a larger fraction of the variance of the underlying ITEs.

- There are formal, decision-theoretic reasons to target the CATE. For example, if a decision maker gets reward $Y_i(d)$ for assigning treatment arm $d$ and faces a cost $C$ per treated unit,

the welfare-maximizing rule is to treat units with $\tau(X_i) > C$, that is, to *threshold the CATE* (Wager, 2025, Ch. 4). This provides a direct link between CATE estimation and optimal treatment assignment (Kitagawa and Tetenov, 2018).

## 1.2 The T-Learner and Regularization Bias

- Given the identification result (2), a natural approach is to estimate $\mu_1(\cdot)$ and $\mu_0(\cdot)$ separately and take their difference.

**Definition 2** (T-Learner)**.** The *T-learner* ("two-model learner") estimates the CATE as

$$\hat{\tau}_T(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x), \tag{3}$$

where $\hat{\mu}_d(\cdot)$ is any nonparametric regression of $Y_i$ on $X_i$ among units with $D_i = d$. Following the nomenclature of Künzel et al. (2019), this is called the T-learner because it fits *two* separate models.

- The T-learner is consistent: if $\hat{\mu}_d(x) \xrightarrow{p} \mu_d(x)$ for each $d$, then $\hat{\tau}_T(x) \xrightarrow{p} \tau(x)$. However, it may not perform well in finite samples due to a phenomenon called *regularization bias*.

- **Regularization bias.** When we fit $\hat{\mu}_0(\cdot)$ and $\hat{\mu}_1(\cdot)$ using methods that involve regularization (e.g., Lasso, ridge, random forests), the amount of regularization is calibrated to predict $Y_i$ accurately within each treatment arm, not to estimate the *difference* $\tau(x)$ accurately. These two objectives require different bias–variance trade-offs:

  - If $\mu_0(x)$ and $\mu_1(x)$ are both complex but their difference $\tau(x)$ is simple (e.g., nearly constant), the T-learner may apply heavy regularization to each surface, and the artifacts from regularization in the two surfaces need not cancel in their difference.
  - In the extreme case where $\tau(x) = 0$ everywhere but the response surfaces oscillate, the T-learner can produce a CATE estimate $\hat{\tau}_T(x)$ that oscillates simply because the two surfaces are regularized differently.

- This concern is particularly acute when the treatment assignment is unbalanced. For instance, if the propensity score is $p(x) = 0.1$ (many more controls than treated units), then $\hat{\mu}_0(\cdot)$ is fit on a much larger sample and can capture more of the oscillation in $\mu_0(x)$, while $\hat{\mu}_1(\cdot)$ is heavily regularized due to the small treated sample. The difference then picks up the oscillation from $\hat{\mu}_0$.

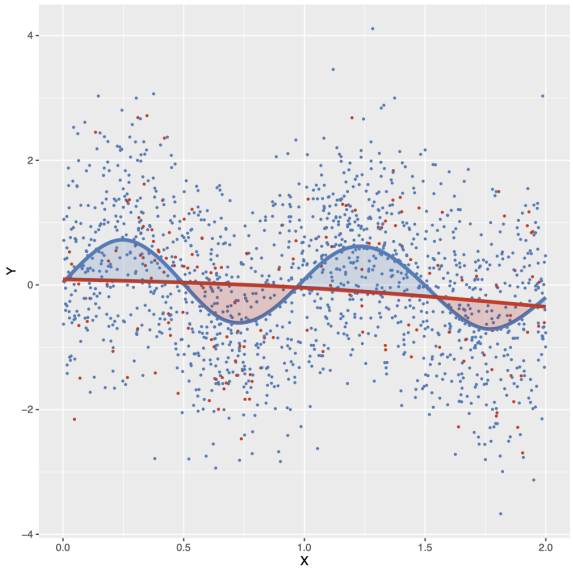


Figure 4.1: Illustration of regularization bias. Both control (blue) and treated (red) units are drawn from the same distribution. Data is generated from an RCT with $\pi = 0.1$, and so there are more controls than treated units. Spline regression learns a more oscillatory model for $\mu_{(0)}(x)$ and a flat one for $\mu_{(1)}(x)$. This results in an oscillatory CATE estimate, illustrated via shading, whereas the true CATE here is identically 0.

Figure 1: Illustration of regularization bias. Source: Wager (2025, Figure 4.1).

- **Regularization-induced confounding.** A second concern arises when the propensity score $p(x)$ varies considerably. In that case, $\hat{\mu}_0(\cdot)$ will be driven by data in areas with more control units (i.e., $p(x)$ close to 0), while $\hat{\mu}_1(\cdot)$ will be driven by areas with more treated units ($p(x)$ close to 1). If there is a covariate shift between the data used to learn $\hat{\mu}_0$ and $\hat{\mu}_1$, their difference may create biases in the CATE estimate.

- These concerns motivate looking for estimators whose loss function is directly calibrated to $\tau(\cdot)$, rather than to the individual response surfaces. This is the central idea behind the semiparametric approach and the R-learner, which we develop next.

## 1.3 Semiparametric Modeling of CATE

- To move toward estimators that directly target $\tau(\cdot)$, it is helpful to start with a *semiparametric* specification.

**The partially linear model.** Suppose the treatment effect heterogeneity is captured by a known

feature map $\psi : \mathcal{X} \to \mathbb{R}^d$ and an unknown coefficient $\beta \in \mathbb{R}^d$:

$$\tau(x) = \psi(x)^\top \beta. \tag{4}$$

For example, if $\mathcal{X}$ contains income and education, one could set

$$\psi(x) = (\text{income in previous year}, \mathbb{1}\{\text{has college degree}\})^\top.$$

- We refer to this as *semiparametric* because the baseline $\mu_0(x)$ and the propensity $p(x)$ are left unrestricted (fully nonparametric), while the treatment effect $\tau(x)$ has a parametric form. Under this specification, the data-generating process becomes a *partially linear model*:

$$Y_i = \mu_0(X_i) + D_i \cdot \psi(X_i)^\top \beta + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i \mid X_i, D_i] = 0. \tag{5}$$

  Estimating the CATE now reduces to estimating the finite-dimensional parameter $\beta$.

**Robinson's decomposition.** The partially linear model (5) was studied by Robinson (1988), who showed that the nuisance function $\mu_0(\cdot)$ can be partialled out. Define the marginal conditional expectation

$$m(x) = \mathbb{E}[Y_i \mid X_i = x] = \mu_0(x) + p(x)\, \psi(x)^\top \beta.$$

Subtracting $m(X_i)$ from both sides of (5) gives the *Robinson decomposition*:

$$\boxed{Y_i - m(X_i) = (D_i - p(X_i))\, \psi(X_i)^\top \beta + \varepsilon_i,} \tag{6}$$

where $\varepsilon_i = Y_i - \mu_0(X_i) - D_i\, \psi(X_i)^\top \beta$ satisfies $\mathbb{E}[\varepsilon_i \mid X_i, D_i] = 0$.

- Equation (6) is a *residual-on-residual regression*: the outcome residual $Y_i - m(X_i)$ is regressed on the treatment residual $(D_i - p(X_i))$ interacted with the features $\psi(X_i)$.

- The key advantage is that the nonparametric nuisance $\mu_0(\cdot)$ has been eliminated. The only unknown objects are $\beta$ (the target) and the two nuisance functions $m(\cdot)$ and $p(\cdot)$, which can be estimated by off-the-shelf ML methods.

- This decomposition is closely related to the Neyman-orthogonal scores introduced in the lecture on debiased inference (DML). The moment condition $\mathbb{E}[\psi(X_i)(Y_i - m(X_i) - (D_i - p(X_i))\psi(X_i)^\top \beta)] = 0$ is Neyman-orthogonal with respect to the nuisance functions $m$ and $p$, which is precisely what enables $\sqrt{n}$-consistent estimation of $\beta$ even when the nuisance functions converge at slower-than-$\sqrt{n}$ rates.

**Estimation with cross-fitting.** In practice, $m(x)$ and $p(x)$ are unknown and must be estimated. Following the DML approach from the debiased inference lecture, we use $K$-fold cross-fitting to avoid overfitting bias:

**Algorithm 1:** Residual-on-Residual Regression for Semiparametric CATE

**Input:** Data $\{(Y_i, D_i, X_i)\}_{i=1}^n$; feature map $\psi(\cdot)$; number of folds $K$.

**Step 1 (Nuisance estimation).** Randomly partition $\{1, \dots, n\}$ into $K$ folds $I_1, \dots, I_K$. For each fold $k$, estimate the nuisance functions using all data *except* fold $k$:

- $\hat{m}^{(-k)}(\cdot)$: regress $Y_i$ on $X_i$ using any ML method (e.g., random forest, Lasso).
- $\hat{p}^{(-k)}(\cdot)$: regress $D_i$ on $X_i$ (e.g., logistic regression, random forest).

**Step 2 (Residuals).** For each observation $i$ in fold $k(i)$, form cross-fit residuals:

$$\widetilde{Y}_i = Y_i - \hat{m}^{(-k(i))}(X_i), \qquad \widetilde{Z}_i = \psi(X_i)\left(D_i - \hat{p}^{(-k(i))}(X_i)\right).$$

**Step 3 (Estimation).** Estimate $\beta$ by OLS of $\widetilde{Y}_i$ on $\widetilde{Z}_i$:

$$\hat{\beta} = \left(\frac{1}{n}\sum_{i=1}^n \widetilde{Z}_i \widetilde{Z}_i^\top\right)^{-1} \frac{1}{n}\sum_{i=1}^n \widetilde{Z}_i \widetilde{Y}_i.$$

**Output:** CATE estimate $\hat{\tau}(x) = \psi(x)^\top \hat{\beta}$.

**Theorem 1** (Asymptotic Normality of $\hat{\beta}$; cf. Wager (2025, Theorem 4.2))**.** *Under the basic setting with SUTVA, unconfoundedness and overlap, suppose that the semiparametric specification (4) holds, that the regression features are bounded $\|\psi(X_i)\|_\infty \le M$, and that we estimate $\beta$ via the $K$-fold cross-fit residual-on-residual regression in Algorithm 1 with nonparametric estimators satisfying the rate conditions: for some constants $\alpha_m \ge 0$, $\alpha_p \ge 1/4$, and $\alpha_m + \alpha_p \ge 1/2$,*

$$n^{2\alpha_m}\frac{1}{|I_k|}\sum_{i:k(i)=k}\left(\hat{m}^{(-k)}(X_i) - m(X_i)\right)^2 \xrightarrow{p} 0, \quad n^{2\alpha_p}\frac{1}{|I_k|}\sum_{i:k(i)=k}\left(\hat{p}^{(-k)}(X_i) - p(X_i)\right)^2 \xrightarrow{p} 0. \tag{7}$$

*Then*

$$\sqrt{n}\left(\hat{\beta} - \beta\right) \xrightarrow{d} \mathcal{N}\left(0, V_\beta\right), \tag{8}$$

*where $V_\beta = \mathrm{Var}\left[\widetilde{Z}_i^*\right]^{-1} \mathbb{E}\left[\left(\varepsilon_i \widetilde{Z}_i^*\right)^{\otimes 2}\right] \mathrm{Var}\left[\widetilde{Z}_i^*\right]^{-1}$ and $\widetilde{Z}_i^* = \psi(X_i)(D_i - p(X_i))$ are the oracle residuals, provided $\mathrm{Var}[\widetilde{Z}_i^*]$ has full rank.*

- The rate conditions require that the nuisance estimators converge at rates whose *product* is faster than $n^{-1/2}$. This is the same product-rate condition that appears in DML: each

nuisance function may converge slower than $n^{-1/4}$ individually, as long as the product of the two rates exceeds $n^{-1/2}$.

- The asymptotic variance $V_\beta$ is the same as if the nuisance functions $m(\cdot)$ and $p(\cdot)$ were known. That is, the first-stage estimation has no asymptotic cost—a consequence of Neyman orthogonality.

**Corollary 2** (Constant Treatment Effect; cf. Wager (2025, Corollary 4.3))**.** *If $\tau(x) = \tau$ for all $x$ (i.e., the treatment effect does not vary with covariates), then (4) holds with $\psi(x) = 1$. The residual-on-residual regression reduces to*

$$\hat{\tau} = \frac{\sum_{i=1}^n \widetilde{Y}_i (D_i - \hat{p}^{(-k(i))}(X_i))}{\sum_{i=1}^n (D_i - \hat{p}^{(-k(i))}(X_i))^2},$$

*and $\sqrt{n}(\hat{\tau} - \tau) \xrightarrow{d} \mathcal{N}(0, V_\tau)$ with*

$$V_\tau = \frac{\mathbb{E}\left[p(X_i)(1-p(X_i))\left((1-p(X_i))\sigma_1^2(X_i) + p(X_i)\sigma_0^2(X_i)\right)\right]}{\mathbb{E}\left[p(X_i)(1-p(X_i))\right]^2},$$

*where $\sigma_d^2(x) = \mathrm{Var}[Y_i(d) \mid X_i = x]$.*

**Remark 1** (Connection to DML)**.** Algorithm 1 is precisely the DML procedure from the debiased inference lecture, applied to the partially linear model (5). The Robinson decomposition (6) provides the Neyman-orthogonal score, and cross-fitting eliminates the overfitting bias from the first-stage ML estimators. When the treatment effect is constant ($\psi(x) = 1$), Corollary 2 recovers the DML estimator for the ATE.

**Remark 2** (Semiparametric efficiency)**.** Under the constant treatment effect model, the AIPW estimator from the causal inference lecture is semiparametrically efficient (i.e., achieves the smallest asymptotic variance among all regular estimators). However, when we impose the additional constraint $\tau(x) = \tau$, estimators that exploit this constraint—such as the residual-on-residual regression—can have *smaller* variance than AIPW (Wager, 2025, Sec. 4.1). This illustrates a general principle: the efficiency of an estimator depends on the assumptions imposed.

# 2 The R-Learner

## 2.1 A Loss Function for Treatment Heterogeneity

- The semiparametric approach of Section 1.3 requires specifying the feature map $\psi(\cdot)$ in advance. What if we do not know which covariates drive treatment heterogeneity, or if the

CATE is a nonlinear function of $X_i$?

- Our goal is to develop an estimator that directly targets the CATE $\tau(\cdot)$ as a function, without committing to a particular parametric form.
- To see how, it is instructive to think about how we moved from linear regression to flexible nonparametric prediction in the supervised learning lecture.

**From prediction to CATE estimation.** In standard prediction, the OLS estimator $\hat{\beta}$ for regressing $Y_i$ on $\psi(X_i)$ can be characterized as the minimizer of the squared-error loss:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{reg}} \left( Y_i;\ \psi(X_i)^\top \beta \right) \right\}, \quad \ell_{\text{reg}}(y;\, z) = (y - z)^2. \tag{9}$$

By the same argument, one can verify that the residual-on-residual regression from Section 1.3 also minimizes a certain least-squares objective. Specifically, define the *R-loss* (for "Robinson loss"):

$$\hat{\ell}^{(-k)}(x,\, y,\, d;\ z) = \left( \left( y - \hat{m}^{(-k)}(x) \right) - \left( d - \hat{p}^{(-k)}(x) \right) z \right)^2. \tag{10}$$

Here $z$ is a candidate value for the treatment effect at covariate value $x$. The cross-fit residual-on-residual estimator from Algorithm 1 is then

$$\hat{\beta} = \arg\min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \hat{\ell}^{(-k(i))} \left( X_i,\, Y_i,\, D_i;\ \psi(X_i)^\top \beta \right) \right\}. \tag{11}$$

- Comparing (9) and (11) makes the parallel between standard prediction and CATE estimation precise. The standard prediction loss $\ell_{\text{reg}}(y;\, z) = (y - z)^2$ evaluates how close the prediction $z$ is to the outcome $y$. The R-loss $\hat{\ell}^{(-k)}(x, y, d;\, z)$ instead evaluates how close $z$ is to the treatment effect: it replaces the raw outcome $y$ with the outcome residual $y - \hat{m}^{(-k)}(x)$ and weights the candidate $z$ by the treatment residual $d - \hat{p}^{(-k)}(x)$.
- One critical difference between (9) and (11) is that, in our setting, the R-loss $\hat{\ell}^{(-k)}$ is *data-dependent*: it takes as input cross-fitted predictions $\hat{m}^{(-k)}(\cdot)$ and $\hat{p}^{(-k)}(\cdot)$. This complicates the theoretical analysis but does not preclude algorithmic development.

**The population R-loss.** To understand why the R-loss targets $\tau(\cdot)$, consider the population version (with true nuisance functions):

$$L(\tau) = \mathbb{E}\left[ \left( (Y_i - m(X_i)) - (D_i - p(X_i))\, \tau(X_i) \right)^2 \right]. \tag{12}$$

By the Robinson decomposition (6), $Y_i - m(X_i) = (D_i - p(X_i))\tau^*(X_i) + \varepsilon_i$ where $\tau^*(\cdot)$ is the true CATE. Substituting and expanding the square,

$$L(\tau) = \mathbb{E}\left[(D_i - p(X_i))^2 \left(\tau^*(X_i) - \tau(X_i)\right)^2\right] + 2\,\mathbb{E}\left[\varepsilon_i (D_i - p(X_i))(\tau^*(X_i) - \tau(X_i))\right] + \mathbb{E}[\varepsilon_i^2].$$

The cross-term vanishes because $\mathbb{E}[\varepsilon_i \mid X_i, D_i] = 0$, and the last term does not depend on $\tau$. Thus:

$$L(\tau) = \mathbb{E}\left[(D_i - p(X_i))^2 \left(\tau(X_i) - \tau^*(X_i)\right)^2\right] + \text{const.}$$

The unconstrained minimizer is $\tau = \tau^*$, confirming that the R-loss is calibrated to the CATE. When minimization is restricted to a function class $\mathcal{F}$, the minimizer is the best approximation to $\tau^*$ within $\mathcal{F}$, measured under the propensity-weighted norm $\mathbb{E}[(D_i - p(X_i))^2(\cdot)^2]$. If $\mathcal{F}$ is rich enough to contain $\tau^*$ (e.g., when $\mathcal{F}$ is the set of all measurable functions), the minimizer coincides with $\tau^*$; otherwise, it is the projection of $\tau^*$ onto $\mathcal{F}$.

**The statistical learning roadmap for CATE.** With the R-loss in hand, we can follow the same progression used in standard supervised learning (Hastie et al., 2009, Chapters 3, 5, 7):

1. **Basis expansion.** Replace the fixed feature map $\psi$ with a growing sequence $\psi : \mathcal{X} \to \mathbb{R}^{d_n}$, where $d_n \to \infty$ with the sample size. For example, $\psi$ could consist of polynomial or trigonometric basis functions with an increasing number of terms. As $d_n$ grows, the model $\tau(x) \approx \psi(x)^\top \beta$ can approximate any reasonable CATE function.

2. **Penalization.** When $d_n$ is large relative to $n$, a direct residual-on-residual regression may be unstable. Introduce a penalty to control complexity. For example, the Lasso penalty on the R-loss gives:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \hat{\ell}^{(-k(i))} \left(X_i,\, Y_i,\, D_i;\, \psi(X_i)^\top \beta\right) + \lambda \sum_{j=1}^{d_n} |\beta_j| \right\}. \tag{13}$$

3. **Tuning.** The penalty parameter $\lambda$ must be chosen in a data-driven way. The simplest approach is to use a validation set and pick $\lambda$ that minimizes the R-loss on that set:

$$\hat{\lambda} = \arg\min_{\lambda} \left\{ \frac{1}{n_{\text{val}}} \sum_{\text{validation}} \hat{\ell}(X_i,\, Y_i,\, D_i;\, \hat{\tau}_\lambda(X_i)) \right\}.$$

   Note that we evaluate the *R-loss*—not the standard prediction loss—on the validation set. This is principled because we are targeting $\tau$, not $Y$. One can also use $K$-fold cross-validation of the R-loss.

4. **Algorithmic methods.** In steps 1–3, the analyst must still specify the basis $\psi(x)$ in advance. Even with a large dictionary and Lasso selection, the quality of the CATE estimate depends on whether the chosen dictionary contains basis elements that capture the true heterogeneity. The final step is to remove this requirement entirely by using algorithmic techniques—decision trees, boosting, neural networks—that *learn the relevant features from the data* as part of the optimization. This leads to the fully nonparametric *R-learner*, which we describe next.

## 2.2 The R-Learner Framework

- The R-learner, introduced by Nie and Wager (2021), extends the residual-on-residual regression to a fully nonparametric setting by allowing the CATE function $\tau(\cdot)$ to belong to an arbitrary function class $\mathcal{F}$.

- **From coefficients to functions.** In the semiparametric approach (Section 1.3) and the penalized R-loss (eq. 13), the optimization is over a coefficient vector $\beta$ given a pre-specified (or pre-selected) basis $\psi(x)$: the CATE is modeled as $\tau(x) = \psi(x)^\top \beta$, and the basis $\psi$ determines *which features* can drive heterogeneity. The R-learner replaces this with optimization over a *function* $\tau \in \mathcal{F}$ directly (eq. 14 below). The function class $\mathcal{F}$ subsumes the role of $\psi$: the ML algorithms, tree partitions, network architecture, kernel structure, implicitly determines how covariates are combined to form the CATE estimate.

- How different algorithms realize this:

  - *Causal forests* partition the covariate space adaptively. Each tree split implicitly selects which variables drive heterogeneity and where the effect changes. The forest weights $\alpha_i(x)$ define a data-adaptive kernel that effectively performs local Robinson regression at each query point $x$. There is no explicit $\psi$—the forest's splitting rules play that role.

  - *Deep neural networks* learn a representation $\phi(x)$ through the hidden layers, and the final layer maps $\phi(x)$ to $\hat{\tau}(x)$. The hidden layers are the learned analog of $\psi$—they perform automatic feature engineering. Crucially, $\phi$ is optimized *jointly* with the output layer by minimizing the R-loss via back-propagation, rather than being fixed in advance.

  - *Lasso on a growing dictionary* is the intermediate case: the analyst provides a large, overcomplete dictionary $\psi(x) \in \mathbb{R}^{d_n}$ (e.g., polynomials and interactions), and the Lasso selects which basis elements are relevant. The basis is still pre-specified, but the selection is data-driven.

**Algorithm 2:** The R-Learner (Nie and Wager, 2021)

**Input:** Data $\{(Y_i, D_i, X_i)\}_{i=1}^n$; function class $\mathcal{F}$; regularizer $\Lambda : \mathcal{F} \to \mathbb{R}_{\geq 0}$; number of folds $K$.

**Step 1 (Nuisance estimation).** Partition $\{1, \ldots, n\}$ into $K$ folds. For each fold $k$, estimate nuisance functions on the out-of-fold data:

$$\hat{m}^{(-k)}(\cdot) \approx \mathbb{E}[Y_i \mid X_i = \cdot], \qquad \hat{p}^{(-k)}(\cdot) \approx \Pr[D_i = 1 \mid X_i = \cdot].$$

Use any ML method (e.g., random forest, boosting, neural network), tuned for optimal predictive accuracy.

**Step 2 (CATE estimation).** Estimate the CATE function by minimizing the empirical R-loss with regularization:

$$\hat{\tau} = \arg\min_{\tau \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left[ \left(Y_i - \hat{m}^{(-k(i))}(X_i)\right) - \left(D_i - \hat{p}^{(-k(i))}(X_i)\right) \tau(X_i) \right]^2 + \Lambda(\tau) \right\}. \quad (14)$$

**Output:** CATE estimate $\hat{\tau}(\cdot)$.

- The R-learner is *modular*: any ML method can be used in Step 1 (for the nuisance functions) and any loss-minimization method can be used in Step 2 (for the CATE function). For example:
  - Step 2 with $\mathcal{F} = \{\psi(\cdot)^\top \beta : \beta \in \mathbb{R}^d\}$ and Lasso penalty $\Lambda(\tau) = \lambda \|\beta\|_1$ gives the penalized semiparametric estimator of Section 2.1.
  - Step 2 with $\mathcal{F}$ being the class of decision trees or forests gives the causal forest (Section 2.3).
  - Step 2 with $\mathcal{F}$ being a neural network class gives a deep learning-based CATE estimator; see Section 2.4.

**Quasi-oracle property.** The key theoretical result is that the R-learner inherits the "robustness" of the Robinson decomposition: errors in the nuisance estimators have only a second-order effect on the CATE estimate.

**Proposition 1** (Quasi-Oracle Property; cf. Nie and Wager (2021))**.** Suppose the nuisance estimators satisfy the rate conditions (7). Then the R-learner $\hat{\tau}$ from Algorithm 2 achieves the same regret bound as the *oracle R-learner* that knows the true nuisance functions $m(\cdot)$ and $p(\cdot)$:

$$\mathbb{E}\left[(D_i - p(X_i))^2 \left(\hat{\tau}(X_i) - \tau(X_i)\right)^2\right] \approx \mathbb{E}\left[(D_i - p(X_i))^2 \left(\hat{\tau}^{\text{oracle}}(X_i) - \tau(X_i)\right)^2\right].$$

That is, the price of not knowing the nuisance functions is asymptotically negligible.

- The quasi-oracle property says that the convergence rate of $\hat{\tau}$ depends only on the complexity of $\tau(\cdot)$ itself, not on the complexity of the nuisance functions $m(\cdot)$ and $p(\cdot)$. This is a powerful result: even if the conditional mean $m(x)$ is very complex and difficult to estimate, the R-learner can recover a smooth CATE $\tau(x)$ at a fast rate.

- Foster and Syrgkanis (2023) provide general guarantees for machine learning with "orthogonal" loss functions that include the R-loss, establishing that the R-learner robustness property extends beyond the semiparametric regime.

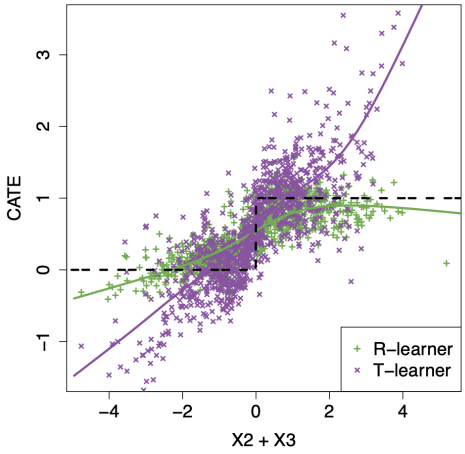


Figure 4.2: Test set CATE estimates generated via the lasso-based R-learner and T-learner. The true CATE function is shown as a black dashed line. The solid lines trace a smooth average of how the CATE estimates vary with $X_2 + X_3$.

Figure 2: Test-set CATE estimates from the lasso-based R-learner and T-learner. Source: Wager (2025, Figure 4.2).

**Remark 3** (Origin of the name)**.** The "R" in R-learner stands for Robinson (1988), who introduced the residual-on-residual regression framework for semiparametric estimation. The R-learner generalizes Robinson's idea from a finite-dimensional $\beta$ to a nonparametric function $\tau(\cdot)$.

**Remark 4** (Other meta-learners)**.** The T-learner (Section 1.2) and the R-learner are two members of a larger family of *meta-learners* for CATE estimation. Other approaches include: the S-learner (a single model for $Y$ with $D$ as a feature), the X-learner (Künzel et al., 2019), and the DR-learner (Kennedy, 2023), which combines doubly robust estimation with flexible nonparametric CATE modeling. See Wager (2025, Ch. 4) for a survey.

## 2.3 Causal Forests

- The R-learner framework tells us *what loss to optimize* (the R-loss), but leaves open the choice of function class $\mathcal{F}$ and optimization method in Step 2. One natural and powerful instantiation is to use random forests. The resulting method, the *causal forest*, has become one of the most widely used tools for CATE estimation.

**Honest causal trees.** Before discussing forests, we start with a single tree. As discussed in the supervised learning lecture, regression trees partition the covariate space into rectangular cells and predict a constant within each cell. For CATE estimation, a "causal tree" should partition the space into cells where the treatment effect is approximately constant, and estimate the within-cell treatment effect.

- **Naive approach.** Grow a standard CART tree for $Y_i$ using $X_i$ as features, then compute $\hat{\tau}(\ell) = \bar{Y}_{1,\ell} - \bar{Y}_{0,\ell}$ in each leaf $\ell$, where $\bar{Y}_{d,\ell}$ is the average outcome among units with $D_i = d$ in leaf $\ell$. The problem is that the tree selects splits to predict $Y_i$ well (i.e., to minimize MSE of $Y$), not to find heterogeneity in $\tau(x)$. Moreover, using the same data to select splits and estimate treatment effects creates selection bias: the tree will find "heterogeneity" that is just noise.

- **Honest estimation.** Athey and Imbens (2016) proposed a key modification: use separate subsets of the data for building the tree structure and for estimating treatment effects within leaves.

**Definition 3** (Honest Estimation)**.** An estimator is *honest* if it partitions the data into two non-overlapping samples:

1. **Training sample $\mathcal{S}^{\text{tr}}$**: used to determine the tree structure (which variables to split on and where).

2. **Estimation sample $\mathcal{S}^{\text{est}}$**: used to estimate the treatment effect within each leaf.

The tree structure from Step 1 is held fixed when computing treatment effect estimates in Step 2.

- **Why honesty matters.** When the same data is used for both selecting the partition and estimating leaf means, the tree algorithm tends to find splits that maximize the *apparent* heterogeneity in treatment effects, even if no true heterogeneity exists. This leads to inflated heterogeneity estimates and undercoverage of confidence intervals. Honest estimation breaks this feedback loop: since the estimation sample is independent of the tree structure, the within-leaf treatment effect estimates $\hat{\tau}(\ell) = \bar{Y}^{\text{est}}_{1,\ell} - \bar{Y}^{\text{est}}_{0,\ell}$ are unbiased.

- **Modified splitting criterion.** For causal trees, the splitting criterion should not minimize the MSE of $Y$ (as in standard CART), but instead maximize the heterogeneity in estimated treatment effects across child nodes. Athey and Imbens (2016) propose criteria that balance between maximizing cross-leaf variation in $\hat{\tau}$ and minimizing within-leaf estimation variance.

- **A disadvantage of sample splitting** is that we lose data: the training sample is not used for estimation, and vice versa. This motivates aggregating many honest trees into a forest.

**Causal forests.** Wager and Athey (2018) developed the causal forest by ensembling many honest causal trees:

- Each tree $b = 1, \ldots, B$ is grown on a random subsample of the data. Following the honesty principle, each tree uses a separate portion of its subsample for building the partition and for estimation.

- The forest prediction at a point $x$ is a weighted average of the training outcomes:

$$\hat{\tau}(x) = \sum_{i=1}^{n} \alpha_i(x) \left(Y_i - \hat{m}(X_i)\right) \cdot \frac{D_i - \hat{p}(X_i)}{\mathbb{E}[\alpha_i(x)(D_i - \hat{p}(X_i))^2]},$$

  where $\alpha_i(x)$ are data-adaptive forest weights reflecting how often observation $i$ falls in the same leaf as $x$ across the $B$ trees. Intuitively, the forest up-weights training observations that are "neighbors" of $x$ according to the tree-based partition.

- The key advantages of forests over individual trees are: (i) averaging reduces variance; (ii) each tree uses a different subsample, so the ensemble explores more of the covariate space; (iii) the forest weights $\alpha_i(x)$ are smooth enough to enable pointwise asymptotic theory.

**Theorem 3** (Asymptotic Normality of Causal Forests; cf. Wager and Athey (2018))**.** *Under regularity conditions (including honesty, subsampling, and a minimum leaf size condition), the causal forest estimator satisfies, for each fixed $x \in \mathcal{X}$,*

$$\frac{\hat{\tau}(x) - \tau(x)}{\hat{\sigma}(x)} \xrightarrow{d} \mathcal{N}(0, 1), \tag{15}$$

*where $\hat{\sigma}^2(x)$ is a consistent variance estimator.*

- **Variance estimation.** Wager and Athey (2018) propose estimating the variance using the *infinitesimal jackknife* (Wager et al., 2014), which computes the sensitivity of the forest prediction to each training observation. This avoids the need to split the data into separate "variance estimation" and "prediction" samples.

- **Confidence intervals.** Theorem 3 enables pointwise confidence intervals for $\tau(x)$:

$$\hat{\tau}(x) \pm z_{\alpha/2}\, \hat{\sigma}(x).$$

  These intervals are valid for a fixed query point $x$ (pointwise coverage). They are *not* simultaneous confidence bands: if one tests many points, some will reject by chance. Honest splitting is essential for the validity of these CIs, as it ensures the asymptotic unbiasedness of $\hat{\tau}(x)$.

**Generalized random forests.** Athey et al. (2019) developed the *generalized random forest* (GRF) framework, which unifies causal forests under a broader methodology. The key idea is to view forests as a way to solve *local moment conditions*: for each query point $x$, estimate $\theta(x)$ by solving

$$\sum_{i=1}^{n} \alpha_i(x) \cdot \psi_{\theta(x),\,\nu(x)}(O_i) = 0, \tag{16}$$

where $\alpha_i(x)$ are data-adaptive forest weights, $O_i = (Y_i, D_i, X_i)$ is the full observation, and $\psi$ is a scoring function (moment condition) that identifies the target parameter.

- For *causal forests*, the moment condition is:

$$\psi_{\tau,\,\mu}(Y, D, X) = \begin{pmatrix} Y - \mu - (D - p(x))\,\tau \\ D - p(x) \end{pmatrix},$$

  where the first component identifies the CATE $\tau(x)$ and the second identifies the local propensity. This connection shows that causal forests can be viewed as local, forest-weighted versions of the Robinson regression.

- **Gradient-based splitting.** Rather than designing a customized splitting criterion for each application, GRF uses a gradient-based approach: at each node, it computes pseudo-outcomes by taking a gradient step on the moment condition (16) and then passes these pseudo-outcomes to a standard CART regression routine. This allows GRF to handle a wide range of estimation problems (quantile regression, instrumental variables, CATE estimation) with a single, efficient implementation.

- **Other applications.** The GRF framework goes beyond causal forests. It can also be used for nonparametric quantile regression, conditional average partial effects, and heterogeneous treatment effects with instrumental variables.

**Remark 5** (Implementation: the `grf` package)**.** The `grf` R package provides a high-performance

implementation of generalized random forests (Athey et al., 2019). A minimal workflow for estimating heterogeneous treatment effects is:

```
library(grf)
cf <- causal_forest(X, Y, W)            # fit causal forest
tau_hat <- predict(cf)$predictions       # CATE estimates
average_treatment_effect(cf)             # forest-based ATE
test_calibration(cf)                     # calibration test
```

The function `causal_forest()` implements honest splitting, subsampling, and the infinitesimal jackknife for variance estimation. The output includes pointwise CATE estimates $\hat{\tau}(X_i)$, standard errors, and an overall ATE estimate with a confidence interval.

## 2.4 Deep Learning for Treatment Heterogeneity

- The methods discussed so far, penalized regression on the R-loss (Section 2.1), causal forests (Section 2.3), use either linear basis expansions or tree-based partitions to model the CATE. Deep neural networks offer a complementary approach that can capture complex nonlinear heterogeneity patterns while scaling to high-dimensional covariates.

**Deep nets as nonparametric estimators.** Farrell et al. (2021) establish the theoretical foundations for using deep ReLU networks in semiparametric inference. Their main contributions are:

- **Nonasymptotic approximation bounds.** For fully connected feedforward networks with ReLU activation $\sigma(x) = \max(x, 0)$, depth $L$ growing with $n$, and width $H$ of the same order of magnitude, the network class $\mathcal{F}_{\text{MLP}}$ can approximate any Hölder-smooth function $f_*$ on $[-1, 1]^d$ with smoothness $\beta > 0$. The key result (their Theorem 1) gives a high-probability bound: for appropriate width $H \asymp n^{d/(2\beta+d)} \log^8 n$ and depth $L \asymp \log n$,

$$\|\hat{f}_{\text{MLP}} - f_*\|_{L_2}^2 = O_P\left(n^{-2\beta/(2\beta+d)} \log^8 n\right),$$

  which matches (up to logarithmic factors) the minimax-optimal rate for nonparametric regression over Hölder balls.

- **Valid semiparametric inference.** These convergence rates are fast enough for second-step semiparametric inference. In particular, when deep nets are used to estimate the nuisance functions $m(\cdot)$ and $p(\cdot)$ in the DML/R-learner framework, the resulting estimator for a finite-dimensional target (e.g., the ATE or a semiparametric CATE coefficient $\beta$) is $\sqrt{n}$-consistent

and asymptotically normal, provided the smoothness condition $\beta/d > 1/4$ holds—the same product-rate requirement as in Theorem 1.

- **Practical implications.** Deep nets can serve as a drop-in replacement for random forests or Lasso in the R-learner (Algorithm 2), with formal theoretical backing. The key advantage is that neural networks can adaptively learn basis functions from the data via back-propagation, rather than requiring the analyst to specify $\psi(x)$ in advance.

**Structural deep learning for individual heterogeneity.** While Farrell et al. (2021) focus on using deep nets as nuisance estimators, Farrell et al. (2020) develop a framework where neural networks are used to learn *heterogeneous parameter functions* within a structural economic model. The key idea is as follows:

- Start with a standard parametric structural model described by a loss function $\ell(\boldsymbol{y}, \boldsymbol{d}, \boldsymbol{\theta})$, where $\boldsymbol{y}$ is the outcome, $\boldsymbol{d}$ is the treatment or policy variable, and $\boldsymbol{\theta} \in \Theta$ is a parameter vector. The structural loss encodes economic restrictions on how outcomes relate to treatments.

- Enrich the model by replacing the fixed parameter $\boldsymbol{\theta}$ with a *parameter function* $\boldsymbol{\theta}(x)$ that varies flexibly with observed covariates $x$. The enriched model is estimated by:

$$\hat{\boldsymbol{\theta}}(\cdot) = \underset{\boldsymbol{\theta} \in \mathcal{F}_{\text{DNN}}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{y}_i,\, \boldsymbol{d}_i,\, \boldsymbol{\theta}(x_i)),$$

  where $\mathcal{F}_{\text{DNN}}$ is a class of deep neural networks.

- The network architecture reflects the model structure: hidden layers map covariates $x$ to a *parameter layer* $\hat{\boldsymbol{\theta}}(x)$, which then enters a *model layer* that computes the structural loss $\ell$. Back-propagation through the model layer ensures that the learned heterogeneity respects the economic structure.

- **Complementarity of structure and ML.** Economic structure provides interpretability, extrapolation, and optimization guarantees (e.g., well-behaved demand curves), while the neural network provides flexibility to capture complex heterogeneity that would be missed by a homogeneous model. Neither alone suffices: ML without structure may produce unreliable counterfactuals, while structure without ML may miss important heterogeneity.

- **Inference via DML.** For second-step quantities of interest, such as average marginal effects, elasticities, or optimal policies, Farrell et al. (2020) derive the influence function for the enriched structural model. This enables valid inference via the DML framework: cross-fit the structural deep learning estimator, form doubly robust scores using the influence function,

and apply standard CLT arguments. Importantly, because the model structure is maintained, the influence function can be computed using ordinary derivatives of the structural loss, automatic differentiation built into neural network software delivers the required quantities without manual derivation.

- **Connection to CATE estimation.** In the special case where $\ell$ is the squared loss and $\boldsymbol{t} = D \in \{0, 1\}$, the structural deep learning framework nests CATE estimation as a special case: the parameter function $\boldsymbol{\theta}(x) = (\mu_0(x), \tau(x))$ captures both the baseline response and the treatment effect. More generally, the framework applies to settings such as heterogeneous demand estimation, personalized pricing, and structural models with individual-level parameters.

**Remark 6** (Deep learning vs. forests for CATE)**.** Both causal forests and deep learning approaches are valid instantiations of the R-learner framework. Causal forests offer built-in inference (via the infinitesimal jackknife), interpretability through variable importance measures, and strong finite-sample performance in moderate-dimensional settings. Deep learning approaches excel with high-dimensional or unstructured covariates (e.g., text, images), can exploit GPU parallelism for large datasets, and—through the structural deep learning framework—can incorporate economic restrictions that tree-based methods cannot. In practice, the choice depends on the application.

# 3 Policy Learning

- So far, we have focused on *estimating* heterogeneous treatment effects. In many applications, however, the ultimate goal is not estimation but *decision making*: we want to learn a rule that tells us whom to treat.

- The problem of learning optimal treatment assignment policies is closely related to, but subtly different from, the problem of estimating treatment heterogeneity. On one hand, policy learning appears easier: all we care about is assigning units to treatment or control, and we do not need to accurately estimate the entire CATE function. On the other hand, any policy we actually want to use must be simple enough to deploy, should not rely on protected characteristics, and may need to respect budget or capacity constraints.

## 3.1 Policy Value and Regret

**Setup.** A treatment assignment *policy* is a mapping

$$\pi : \mathcal{X} \to \{0, 1\}, \tag{17}$$

such that individuals with covariates $X_i = x$ are treated if and only if $\pi(x) = 1$. Under the potential outcome framework, the expected outcome when treatment is assigned according to policy $\pi$ is the *policy value*:

$$V(\pi) = \mathbb{E}\left[Y_i(\pi(X_i))\right]. \tag{18}$$

The decision maker wants to use data to learn a policy $\hat{\pi}$ such that $V(\hat{\pi})$ is large.

- **Connection to the CATE.** Using $Y_i(\pi(x)) = Y_i(0) + \pi(x)\left(Y_i(1) - Y_i(0)\right)$, we can decompose the value as

  $$V(\pi) = \mathbb{E}[Y_i(0)] + \mathbb{E}\left[\pi(X_i)\,\tau(X_i)\right].$$

  The first term does not depend on the policy. Thus, comparing two policies reduces to comparing $\mathbb{E}[\pi(X_i)\,\tau(X_i)]$. If there are no constraints on $\pi$, the optimal unrestricted policy simply thresholds the CATE:

  $$\pi^*(x) = \mathbb{1}\{\tau(x) > 0\}. \tag{19}$$

- **Plug-in approach.** One could first estimate $\hat{\tau}(\cdot)$ using methods from the previous sections and then set $\hat{\pi}(x) = \mathbb{1}\{\hat{\tau}(x) > 0\}$. This may be reasonable in some applications, but it can produce policies that are hard to interpret or that do not respect practical constraints (e.g., budget limits, fairness requirements, simplicity).

- **Restricted policy classes.** In practice, the policymaker is often constrained to choose a policy $\pi$ from some class $\Pi$ of acceptable policies. For example, $\Pi$ may encode restrictions on the functional form (e.g., linear eligibility rules, depth-limited decision trees), which variables the policy is allowed to use, or capacity constraints. Simple examples include:

  - Linear thresholding rules: $\pi(x) = \mathbb{1}\{a \cdot x \geq c\}$ for some vector $a$ and threshold $c$.
  - Fixed-depth decision trees.

**Definition 4** (Regret)**.** Given a policy class $\Pi$, the *optimal policy* and the *regret* of any policy $\pi \in \Pi$ are

$$\pi^* \in \arg\max\left\{V(\pi') : \pi' \in \Pi\right\}, \qquad R(\pi) = \sup_{\pi' \in \Pi} V(\pi') - V(\pi). \tag{20}$$

The regret measures how much value is lost by deploying $\pi$ instead of the best policy in $\Pi$.

- The goal of policy learning is to find $\hat{\pi} \in \Pi$ with small regret $R(\hat{\pi})$. This is a *learning* task (rather than estimation): we do not require $\hat{\pi}$ to converge to $\pi^*$ in functional form, only that $R(\hat{\pi}) \to 0$.

## 3.2 Policy Evaluation

- Before learning a policy, we need a way to *evaluate* one: given a candidate policy $\pi$, how can we estimate its value $V(\pi)$ from observational data?

**Inverse-propensity weighting.** Under unconfoundedness, we can estimate $V(\pi)$ via IPW:

$$\widehat{V}_{\mathrm{IPW}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathbb{1}\{D_i = \pi(X_i)\}}{D_i\, p(X_i) + (1-D_i)(1-p(X_i))}\, Y_i. \tag{21}$$

This averages outcomes over those observations whose treatment matches the policy prescription $\pi(X_i)$, using inverse-propensity weighting to correct for the fact that some relevant potential outcomes are unobserved.

**AIPW for policy evaluation.** As with ATE estimation, IPW is generally inefficient and sensitive to estimation error in $p(x)$. A more robust alternative is the *augmented inverse-propensity weighted* (AIPW) estimator. Define the doubly robust scores

$$\widehat{\Gamma}_i = \hat{\mu}_1(X_i) - \hat{\mu}_0(X_i) + \frac{D_i - \hat{p}(X_i)}{\hat{p}(X_i)(1-\hat{p}(X_i))}\left(Y_i - \hat{\mu}_{D_i}(X_i)\right), \tag{22}$$

which are doubly robust pseudo-outcomes that estimate the CATE $\tau(X_i)$, with nuisance functions estimated via cross-fitting. A welfare-equivalent objective that is especially convenient for analysis is

$$\widehat{A}_{\mathrm{AIPW}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} \left(2\pi(X_i) - 1\right)\widehat{\Gamma}_i. \tag{23}$$

- $\widehat{A}_{\mathrm{AIPW}}(\pi) = 2\,\widehat{V}_{\mathrm{AIPW}}(\pi) + C$, where $\widehat{V}_{\mathrm{AIPW}}(\pi) = \frac{1}{n}\sum_i \big[\pi(X_i)\,\widehat{\Gamma}_i^{(1)} + (1-\pi(X_i))\,\widehat{\Gamma}_i^{(0)}\big]$ is the standard AIPW value estimator (with $\widehat{\Gamma}_i^{(w)}$ the arm-$w$ AIPW scores, so that $\widehat{\Gamma}_i = \widehat{\Gamma}_i^{(1)} - \widehat{\Gamma}_i^{(0)}$) and $C = -\frac{1}{n}\sum_i \big[\widehat{\Gamma}_i^{(0)} + \widehat{\Gamma}_i^{(1)}\big]$ does not depend on $\pi$. Hence a policy is a maximizer of $\widehat{V}_{\mathrm{AIPW}}(\pi)$ if and only if it is a maximizer of $\widehat{A}_{\mathrm{AIPW}}(\pi)$, and the $\widehat{A}_{\mathrm{AIPW}}$ formulation is more convenient for both computation and theory.

- **Policy comparison.** To compare two policies $\pi_1$ and $\pi_2$, we can estimate

$$\widehat{\Delta}_{\mathrm{AIPW}}(\pi_1, \pi_2) = \widehat{V}_{\mathrm{AIPW}}(\pi_1) - \widehat{V}_{\mathrm{AIPW}}(\pi_2) = \frac{1}{n}\sum_{i=1}^{n} \left(\pi_1(X_i) - \pi_2(X_i)\right)\widehat{\Gamma}_i.$$

  When $\pi_1$ and $\pi_2$ agree on the action to take for most units, $\widehat{\Delta}_{\mathrm{AIPW}}$ only averages scores in the region where the policies differ, yielding a considerable improvement in precision.

## 3.3 Empirical Welfare Maximization

- We now turn to the task of *learning* a good policy $\hat{\pi}(\cdot)$ from data. The *empirical welfare maximization* (EWM) approach, introduced by Kitagawa and Tetenov (2018), selects the policy that maximizes an estimated value function over the class $\Pi$:

$$\hat{\pi} = \arg\max \left\{ \widehat{V}(\pi) : \pi \in \Pi \right\}, \tag{24}$$

  where $\widehat{V}(\pi)$ can be either the IPW or AIPW value estimator.

**Regret bounds.** The key theoretical question is: how quickly does $R(\hat{\pi}) \to 0$? The answer depends on the complexity of the policy class $\Pi$ and the quality of the value estimator $\widehat{V}$.

- A standard argument shows that the regret of any EWM policy satisfies

$$R(\hat{\pi}) \leq 2 \sup_{\pi \in \Pi} \left| \widehat{V}(\pi) - V(\pi) \right|. \tag{25}$$

  Thus, proving low regret reduces to proving *uniform* convergence of $\widehat{V}(\pi)$ to $V(\pi)$ over $\pi \in \Pi$.

- The complexity of $\Pi$ is measured by its *Vapnik–Chervonenkis (VC) dimension* $\mathrm{VC}(\Pi)$. Intuitively, $\mathrm{VC}(\Pi)$ captures the number of parameters needed to specify a policy in $\Pi$. Examples:
  - Linear decision rules $\pi(x) = \mathbb{1}\{a \cdot x \geq c\}$ in $\mathbb{R}^p$: $\mathrm{VC}(\Pi) = p + 1$.
  - Depth-$L$ decision trees on $\mathbb{R}^p$: $\mathrm{VC}(\Pi) = \mathcal{O}(2^L \log(p))$.

**EWM with known propensity scores.** Kitagawa and Tetenov (2018) study the case where the propensity score $p(x)$ is known (as in a randomized experiment). They show that the IPW-based EWM rule achieves regret

$$R(\hat{\pi}_{\mathrm{IPW}}) = O_P\left( \sqrt{\frac{\mathrm{VC}(\Pi)}{n}} \right), \tag{26}$$

and that this rate is *minimax optimal*: no policy learning algorithm can achieve a uniformly faster regret rate over a minimally constrained class of data-generating distributions.

**EWM with estimated nuisance functions.** Athey and Wager (2021) extend this framework to observational settings where the propensity score and conditional response surfaces are unknown. Using the AIPW scores $\widehat{\Gamma}_i$ from (22) with cross-fitting, they learn $\hat{\pi}$ by maximizing the AIPW advantage:

$$\hat{\pi}_{\mathrm{AIPW}} = \arg\max_{\pi \in \Pi} \left\{ \frac{1}{n} \sum_{i=1}^{n} (2\pi(X_i) - 1)\, \widehat{\Gamma}_i \right\}. \tag{27}$$

**Theorem 4** (Regret Bound for AIPW-Based EWM; cf. Athey and Wager (2021, Theorem 1))**.** *Under regularity conditions (unconfoundedness, overlap, bounded outcomes, and nuisance estimators satisfying product-rate conditions as in DML), and assuming* $\mathrm{VC}(\Pi_n) \leq n^\beta$ *for some* $\beta < 1/2$*, the AIPW-based EWM policy satisfies*

$$\limsup_{n\to\infty} \sqrt{n}\, \mathbb{E}\left[R(\hat{\pi}_{\mathrm{AIPW}})\right] \leq C\sqrt{\mathrm{VC}(\Pi)\left(\mathrm{Var}[\tau(X_i)] + \mathbb{E}\left[\frac{\sigma_0^2(X_i)}{1-p(X_i)} + \frac{\sigma_1^2(X_i)}{p(X_i)}\right]\right)}, \tag{28}$$

*where* $C$ *is a universal constant.*

- The regret bound scales with $\sqrt{\mathrm{VC}(\Pi)/n}$: richer policy classes are harder to learn over. It also scales with the variance of the AIPW scores, which reflects the difficulty of the underlying causal inference problem.
- This result achieves the same $1/\sqrt{n}$ rate as Kitagawa and Tetenov (2018), but in a more general setting where propensity scores are unknown and estimated via ML. The doubly robust scores are crucial: they ensure that first-stage estimation errors have only a second-order effect on the regret.

## 3.4 Policy Learning as Weighted Classification

- A key practical insight is that empirical welfare maximization can be reformulated as a *weighted classification* problem (Wager, 2025, Ch. 5).
- From (27), the learned policy maximizes $\frac{1}{n}\sum_{i=1}^n (2\pi(X_i)-1)\widehat{\Gamma}_i$. This is equivalent to the weighted classification objective:

$$\hat{\pi}_{\mathrm{AIPW}} = \arg\max_{\pi\in\Pi} \left\{ \frac{1}{n}\sum_{i=1}^n \underbrace{(2\pi(X_i)-1)\ \mathrm{sgn}(\widehat{\Gamma}_i)}_{\text{classification objective}} \cdot \underbrace{|\widehat{\Gamma}_i|}_{\text{sample weight}} \right\}. \tag{29}$$

  In words: the policy tries to assign $\pi(X_i) = 1$ when $\widehat{\Gamma}_i > 0$ (estimated positive treatment effect) and $\pi(X_i) = 0$ when $\widehat{\Gamma}_i < 0$, with each observation weighted by $|\widehat{\Gamma}_i|$ (the magnitude of the estimated effect).
- This means we can use any software for weighted classification (e.g., SVMs, boosted trees, logistic regression) to solve the EWM problem.
- **Caveat.** The weighted classification formulation is useful computationally, but one should not over-interpret it. In typical settings, the signs of $\widehat{\Gamma}_i$ are noisy, and even the optimal policy

$\pi^*$ will "misclassify" many observations. Standard classification diagnostics (e.g., accuracy) can be misleading.

**Remark 7** (The role of the policy class $\Pi$)**.** Throughout this section, we sought to learn the best policy *within* a restricted class $\Pi$, which typically excludes the unrestricted optimum $\pi^*_{\text{unrestr.}}(x) = \mathbb{1}\{\tau(x) > 0\}$. This restriction is often a feature rather than a limitation, because the covariates $X_i$ play two distinct roles in policy learning: (i) *identification* by achieving unconfoundedness, for which we want a rich set of covariates entering flexibly; and (ii) *deployment* by defining the rule $\pi(\cdot)$ that will be acted upon, for which we may prefer a simple function of a few interpretable variables. Accordingly, $\Pi$ should be chosen to contain only policies that are feasible to deploy, e.g., those that do not depend on variables that are difficult to measure, gameable, or legally protected (Wager, 2025, Ch. 5).

`Zhan Gao, April 20, 2026`

# References


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

Athey, S., J. Tibshirani, and S. Wager (2019). Generalized random forests. *The Annals of Statistics 47*(2), 1148–1178.

Athey, S. and S. Wager (2021). Policy learning with observational data. *Econometrica 89*(1), 133–161.

Farrell, M. H., T. Liang, and S. Misra (2020). Deep learning for individual heterogeneity: An automatic inference framework. *arXiv preprint arXiv:2010.14694*.

Farrell, M. H., T. Liang, and S. Misra (2021). Deep neural networks for estimation and inference. *Econometrica 89*(1), 181–213.

Foster, D. J. and V. Syrgkanis (2023). Orthogonal statistical learning. *The Annals of Statistics 51*(3), 879–908.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning*. Springer New York.

Kennedy, E. H. (2023). Towards optimal doubly robust estimation of heterogeneous causal effects. *Electronic Journal of Statistics 17*(2), 3008–3049.

Kitagawa, T. and A. Tetenov (2018). Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica 86*(2), 591–616.

Künzel, S. R., J. S. Sekhon, P. J. Bickel, and B. Yu (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences 116*(10), 4156–4165.

Nie, X. and S. Wager (2021). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika 108*(2), 299–319.

Robinson, P. M. (1988). Root-N-consistent semiparametric regression. *Econometrica 56*(4), 931–954.

Wager, S. (2025). *Causal inference: A statistical learning approach*.

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

Wager, S., T. Hastie, and B. Efron (2014). Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research 15*(1), 1625–1651.