# Nonparametric Estimation of Heterogeneous Treatment Effects: From Theory to Learning Algorithms
# Supplementary Materials

**Alicia Curth**
University of Cambridge
amc253@cam.ac.uk

**Mihaela van der Schaar**
University of Cambridge, UCLA & The Alan Turing Institute
mv472@cam.ac.uk

## 1 ASSUMPTIONS AND ADDITIONAL ANALYSES

In this section, we revisit the assumptions made in Section 4. First, we discuss estimation under assumed sparsity instead of smoothness, and then we discuss the assumptions associated with Kennedy (2020)'s theorem on pseudo-outcome regression.

### 1.1 Additional Analyses on Minimax Performance Using Assumptions on Sparsity

**Why consider minimax error rates?** In the main text, we relied on assumptions of smoothness on underlying functions and below we discuss estimation under assumed sparsity. We do so to nonparametrically *quantify* the hardness of the different estimation problems, allowing us to systematically compare the minimax performance of different learners. The remainders derived in Theorem 1 allow for much more general analyses than what we discuss here, and could be used to assess the relative performance of the different learners using *any* assumption on the learning rates associated with the "difficulty" of the functions $\tau(x)$, $\mu_w(x)$ and $\pi(x)$. We rely on smoothness and sparsity due to their intuitive appeal, generality and usage in related work (Alaa and van der Schaar, 2018; Kennedy, 2020).

**Minimax rates for estimation under sparsity** Previously, we relied on assumed smoothness of the different regression functions, to illustrate the effect of differences in underlying complexity of the nuisance functions. Instead of smoothness only, we now consider functions with sparsity and additive sparsity as defined in assumption M3 in Yang et al. (2015), which is often a necessary assumption enabling estimation when data is high-dimensional (e.g. $d > n$, where $X \in \mathbb{R}^d$). A function $f$ satisfies (additive) sparsity if it depends on $d^* \asymp min(n^\gamma, d)$ variables for some $\gamma \in (0, 1)$ but admits an additive structure $f = \sum_{s=1}^k f_s$ where each of the $k$ component functions $f_s$ depends on a small number of predictors ($d_s$). As special cases of this assumption we have both the more standard sparsity assumption, where one $f$ depends on one small subset $d^* \leq min(n, d)$ of the predictors (i.e. $k = 1$), as well as the case where $f$ is completely additive ($d_s = 1$ for all $s$).

For ease of exposure, we also assume that all additive components $f_s$ have the same smoothness $p_s = p$, dimension $d_s = d^*$ and magnitude. As shown by Yang et al. (2015), this leads to the minimax rate $kn^{-2p/(2p+d^*)} + k\frac{d^* \log(d/d^*)}{n}$ in squared error of estimation of $f$. The first term is Stone (1980)'s nonparametric minimax rate for a p-smooth function as we considered in the main text, but with $d^* \leq d$ known, while the second term captures the uncertainty in variable selection.

**Learner performance under sparsity** We can use this minimax rate to compare the error rates attained by the different learners similarly as we used smoothness in the main text. For example, if we assume that each regression function $f$ is $d^*$-sparse and linear in $X$, we have the minimax rate $\frac{d^* \log(d/d^*)}{n}$ for an estimator $\hat{f}$ (Raskutti et al., 2009), and the squared error of estimation using the lasso can attain the minimax rate $\frac{d^* log(d)}{n}$ (Bickel et al., 2009).

**Assumption.** *Assume that $\tau(x)$, $\mu_w(x)$ and $\pi(x)$ are linear in $x$ and $d_\tau$, $d_{\mu_w}$ and $d_\pi$-sparse, respectively, and each function $f$ can hence be estimated with squared error rate $\frac{d_f log(d)}{n}$.*

Then, we immediately have the following corollary:

**Corollary.** *Using Theorem 1, we have the following error rates on the four learners under the sparsity assumptions discussed above:*

- *For the plug-in learner:*
$$\mathbb{E}[(\hat{\tau}_{plug,\hat{\eta}}(x) - \tau(x))^2] \lesssim \frac{(d_{\mu_0} + d_{\mu_1}) log(d)}{n} \tag{1}$$
- *For the RA-learner:*
$$\mathbb{E}[(\hat{\tau}_{RA,\hat{\eta}}(x) - \tau(x))^2] \lesssim \frac{d_\tau log(d)}{n} + \frac{(d_{\mu_0} + d_{\mu_1}) log(d)}{n} \tag{2}$$
- *For the PW-learner:*
$$\mathbb{E}[(\hat{\tau}_{PW,\hat{\eta}}(x) - \tau(x))^2] \lesssim \frac{d_\tau log(d)}{n} + \frac{d_\pi log(d)}{n} \tag{3}$$
- *For the DR-learner:*
$$\mathbb{E}[(\hat{\tau}_{DR,\hat{\eta}}(x) - \tau(x))^2] \lesssim \frac{d_\tau log(d)}{n} + \frac{(d_{\mu_0} + d_{\mu_1}) d_\pi log^2(d)}{n^2} \tag{4}$$

This leads to analogous conclusions on learner performance as presented in the main text, but based on sparsity instead of smoothness. For example, two-step learners can outperform the plug-in learner if $d_\tau < max(d_{\mu_1}, d_{\mu_0})$, i.e. if the treatment effect depends on less covariates than each potential outcome function. This would *not* be the case if $\mu_1(x)$ and $\mu_0(x)$ would depend on completely disjoint sets of covariates, as in setting (iii) in our experiments. Further, if $d_\tau < max(d_{\mu_1}, d_{\mu_0})$, then RA-learner and plug-in learner are expected to perform equally, and PW- outperforms RA-learner if $d_\pi < max(d_{\mu_1}, d_{\mu_0})$. Finally, the DR-learner attains the oracle rate for CATE estimation if $d_\tau > \frac{(d_{\mu_0}+d_{\mu_1}) d_\pi log(d)}{n}$ (this is shown also in Kennedy (2020)).

Using the more general formulation of Yang et al. (2015), relying not on linear but general sparse functions, would allow to consider even more general scenarios and derive conditions under which each learner outperforms the others based on smoothness of nonlinear functions *and* sparsity.

**Further comparison with existing results** As previously mentioned, some of the different learners have been discussed separately in related work. In particular, asymptotic analyses for the DR-learner under smoothness and sparsity assumptions were presented in Kennedy (2020), and Alaa and van der Schaar (2018) derive an error bound for Bayesian estimation of treatment effects using plug-in learners, based on an information-theoretic approach assuming smoothness and sparsity, which gives results for the plug-in learner analogous to ours. Yet, by analyzing the four main strategies for nonparametric meta-learning of CATE within one coherent framework, we contribute to existing work by building systematic understanding of the relative strengths and weaknesses of different estimation strategies in different scenarios. In particular, theoretical comparisons between one- and two-step learners often emphasize the favorable properties of two-step learners (e.g. for the X-learner in Künzel et al. (2019) and the DR-learner in Kennedy (2020)) because it is typically assumed that CATE is much simpler than the baseline outcome function $\mu_0(x)$ (Künzel et al., 2019). This assumption, however, is not always reflected in the DGPs used to evaluate performance of CATE estimators elsewhere. The IHDP benchmark used in (among others) Shalit et al. (2017); Shi et al. (2019); Hassanpour and Greiner (2020); Wu et al. (2020), based on Hill (2011)'s simulation setting B (discussed in detail below), does *not* satisfy this assumption, leading to conclusions based on empirical performance that seemingly stand in conflict with theoretical analyses highlighting mainly the favorable properties of two-step learners.

### 1.2 Assumptions on Estimators for Pseudo-outcome Regression

To be able to bound the error in pseudo-outcome regression using Kennedy (2020)'s Theorem 1 (which leads to the additive error decomposition using the oracle rate of estimation of CATE) we need a mild assumption on the second-stage regression model necessary to ensure stability of the second stage regression (Kennedy, 2020):

**Assumption.** ***Regularity of regression estimators***
*We need two mild assumptions on the regularity of our second-stage regression estimators $\hat{\mathbb{E}}_n$. $\hat{\mathbb{E}}_n$ needs to satisfy that:*

1. *$\hat{\mathbb{E}}_n(Y|X=x)+c=\hat{\mathbb{E}}_n(Y+c|X=x)$ for any constant $c$*
2. *If $\mathbb{E}[Y|X=x]=\mathbb{E}[W|X=x]$ then*

$$\mathbb{E}\left[\{\hat{\mathbb{E}}_n[W|X=x]-\mathbb{E}[W|X=x]\}^2\right] \asymp \mathbb{E}\left[\{\hat{\mathbb{E}}_n[Y|X=x]-\mathbb{E}[Y|X=x]\}^2\right]$$

The first assumption enforces that adding a constant to an outcome pre- or post-regression leads to the same result, whereas the second assumption says that the regression method results in the same error (up to constants) for two variables with the same conditional means, regardless of e.g. variance (Kennedy, 2020).

# 2 PROOFS

## 2.1 Proof of Theorem 1

Here, we consider the term $\mathbb{E}[(\mathbb{E}[\tilde{Y}_{\hat{\eta}}(x)|X=x,\mathcal{D}_0]-\tau(x))^2]$ in detail for each learner. For the DR-learner, this was proven in Kennedy (2020), but we restate the proof here for completeness. By the tower property, we have for the term $R=\mathbb{E}[\tilde{Y}_{\hat{\eta}}(x)|X=x,\mathcal{D}_0]-\tau(x)$ for each two-step learner:
(1) RA-Learner:

$$\begin{aligned}R&=\pi(x)[\mu_1(x)-\hat{\mu}_0(x)]+(1-\pi(x))[\hat{\mu}_1(x)-\mu_0(x)]-(\mu_1(x)-\mu_0(x))\\&=\pi(x)[\mu_0(x)-\hat{\mu}_0(x)]+(1-\pi(x))[\hat{\mu}_1(x)-\mu_1(x)]\end{aligned}$$

(2) PW-Learner:

$$\begin{aligned}R&=\frac{\pi(x)}{\hat{\pi}(x)}\mu_1(x)-\frac{1-\pi(x)}{1-\hat{\pi}(x)}\mu_0(x)-(\mu_1(x)-\mu_0(x))\\&=(\frac{\pi(x)}{\hat{\pi}(x)}-1)\mu_1(x)-(\frac{1-\pi(x)}{1-\hat{\pi}(x)}-1)\mu_0(x)\\&=\frac{1}{\hat{\pi}(x)}(\pi(x)-\hat{\pi}(x))\mu_1(x)-\frac{1}{1-\hat{\pi}(x)}(\hat{\pi}(x)-\pi(x))\mu_0(x)\end{aligned}$$

(3) DR-Learner:

$$R=\frac{1}{\hat{\pi}(x)}(\pi(x)-\hat{\pi}(x))(\hat{\mu}_1(x)-\mu_1(x))-\frac{1}{1-\hat{\pi}(x)}(\hat{\pi}(x)-\pi(x))(\hat{\mu}_0(x)-\mu_0(x))$$

Using the identity $(a+b)^2\leq 2(a^2+b^2)$ and assumption 2, this yields for the square $R^2=(\mathbb{E}[\tilde{Y}_{\hat{\eta}}(x)|X=x,\mathcal{D}_0]-\tau(x))^2$ for the
(1) RA-learner:

$$\begin{aligned}R^2&\leq 2(\pi(x)^2[\mu_0(x)-\hat{\mu}_0(x)]^2+(1-\pi(x))^2[\hat{\mu}_1(x)-\mu_1(x)]^2)\\&\leq 2(1-\omega)^2(\sum_{w\in\{0,1\}}(\hat{\mu}_w(x)-\mu_w(x))^2)\end{aligned}$$

(2) PW-learner:

$$R^2\leq 2(\frac{\mu_1^2(x)}{\hat{\pi}^2(x)}+\frac{\mu_0^2(x)}{(1-\hat{\pi}(x))^2})(\hat{\pi}(x)-\pi(x))^2\leq\frac{4C^2}{\delta^2}(\hat{\pi}(x)-\pi(x))^2$$

(3) DR-learner:

$$\begin{aligned}R^2&\leq 2(\hat{\pi}(x)-\pi(x))^2(\frac{(\hat{\mu}_1(x)-\mu_1(x))^2}{\hat{\pi}^2(x)}+\frac{(\hat{\mu}_0(x)-\mu_0(x))^2}{(1-\hat{\pi})^2})\\&\leq\frac{2}{\delta^2}(\hat{\pi}(x)-\pi(x))^2(\sum_{w\in\{0,1\}}(\hat{\mu}_w(x)-\mu_w(x))^2)\end{aligned}$$

The theorem follows by taking expectations over $R^2$, and applying assumption 3.

### 2.2 Proof of equation 5

In the following, we denote by $p(\cdot)$ the pdf of $\mathbb{P}$. Further, we let $R_1 = \mathbb{E}_{X\sim\mathbb{P}(\cdot|W=1)}[(\hat{\mu}_1(X) - \mu_1(X))^2]$. By repeated application of Bayes rule and the law of total probability we can show that:

$$\mathbb{E}_{X\sim\mathbb{P}(\cdot)}[(\hat{\mu}_1(x) - \mu_1(x))^2] = \int (\hat{\mu}_1(x) - \mu_1(x))^2 p(x) dx$$

$$= \mathbb{P}(W=1) \int (\hat{\mu}_1(x) - \mu_1(x))^2 p(x|W=1)dx + (1-\mathbb{P}(W=1)) \int (\hat{\mu}_1(x) - \mu_1(x))^2 p(x|W=0)dx$$

$$= \mathbb{P}(W=1)R_1 + (1-\mathbb{P}(W=1)) \int (\hat{\mu}_1(x) - \mu_1(x))^2 \frac{p(x|W=1)}{p(x|W=1)} p(x|W=0)dx$$

$$= \mathbb{P}(W=1)R_1 + (1-\mathbb{P}(W=1)) \int (\hat{\mu}_1(x) - \mu_1(x))^2 \frac{\frac{\mathbb{P}(W=0|x)p(x)}{(1-\mathbb{P}(W=1))}}{\frac{\mathbb{P}(W=1|x)p(x)}{\mathbb{P}(W=1)}} p(x|W=1)dx$$

$$= \mathbb{P}(W=1)R_1 + \mathbb{P}(W=1) \int (\hat{\mu}_1(x) - \mu_1(x))^2 \frac{\mathbb{P}(W=0|x)}{\mathbb{P}(W=1|x)} p(x|W=1)dx$$

$$= \mathbb{P}(W=1) \int \left(1 + \frac{\mathbb{P}(W=0|x)}{\mathbb{P}(W=1|x)}\right) (\hat{\mu}_1(x) - \mu_1(x))^2 p(x|W=1)dx$$

$$= \mathbb{E}_{X\sim\mathbb{P}(\cdot|W=1)} \left[\mathbb{P}(W=1) \left(1 + \frac{1-\pi(X)}{\pi(X)}\right) (\hat{\mu}_1(X) - \mu_1(X))^2\right]$$

# 3 LEARNING ALGORITHMS AND IMPLEMENTATION

In this section we first give pseudo code for the two-step learners, then discuss the loss functions associated with the different SNets, and finally discuss implementation details.

### 3.1 Two-step learner pseudo code

Below, we present the pseudo code for the two-step learners. As discussed in Section 5, we used no form of sample splitting (option 3 in the algorithm described below) in our experiments, yet both cross-fitting (option 1) and a single sample split (option 2) could be used to implement a two-step learner for which the theoretical guarantees hold as analysed.

**Algorithm:** Two-step learner

```
Inputs: A sample $\mathcal{D} = \{Y_i, W_i, X_i\}_{i=1}^n$, a learning algorithm $\mathcal{A}$, a first-stage fitting strategy and a second stage pseudo-outcome formula $\tilde{Y}_{\hat{\eta}} = f_{\tilde{Y}}(Y, W, X; \hat{\eta})$
First stage: nuisance model estimation
if fitting strategy is cross-fitting then
    split the sample $\mathcal{D}$ in $k$ non-overlapping folds
    for $k \leftarrow 1 : K$ do
        Fit nuisance models $\hat{\eta}_{-k} = \mathcal{A}(\mathcal{D}_{-k})$ on all but the $k^{th}$ fold
        Predict $\tilde{Y}_i = f_{\tilde{Y}}(Y_i, W_i, X_i; \hat{\eta}_{-k})$ for $i \in \mathcal{D}_k$ using the nuisance model $\hat{\eta}_{-k}$
    end for
else if Fitting strategy is sample splitting then
    split the sample into $\mathcal{D}_1$ and $\mathcal{D}_2$
    Fit nuisance model $\hat{\eta} = \mathcal{A}(\mathcal{D}_1)$ on $\mathcal{D}_1$
    Predict $\tilde{Y}_i = f_{\tilde{Y}}(Y_i, W_i, X_i; \hat{\eta})$ for $i \in \mathcal{D}_2$ using the nuisance model $\hat{\eta}$
else
    Fit nuisance model $\hat{\eta} = \mathcal{A}(\mathcal{D})$ on full sample
    Predict $\tilde{Y}_i = f_{\tilde{Y}}(Y_i, W_i, X_i; \hat{\eta})$ for $i \in \mathcal{D}$ using the nuisance model $\hat{\eta}$
end if
```

```
Second stage: CATE estimation
estimate $\tau(x)$ as a function of $x$ by regressing $\{\tilde{Y}_i\}$ on $\{X_i\}$ as $\hat{\tau}(x) = \mathcal{A}(\{\tilde{Y}_i, X_i\})$
Output: $\hat{\tau}(x)$
```

### 3.2 Loss functions for SNets

In this section we present the loss functions we use to implement all SNet variants. For SNets 1 - 3, these are adapted from Shalit et al. (2017), Shi et al. (2019) and Hassanpour and Greiner (2020), respectively, but not exactly identical – we did not use re-weighting, re-balancing or TMLE-regularization schemes in estimating nuisance parameters, as we wish to consider only direct plug-in estimators. Strictly speaking, we therefore adapted only their model architectures. Further, it would be possible to assign different weights to different loss components (e.g. loss on propensity score estimation) below, however, we do not do so here to avoid adding additional hyper-parameters.

**SNet-1 (TARNet, adapted from Shalit et al. (2017))**

$$\min_{h_0, h_1, \Phi} \frac{1}{n} \sum_{i=1}^{n} L(h_{W_i}(\Phi(X_i)), Y_i) + \lambda \mathcal{R}(h_0, h_1) \tag{5}$$

where $\Phi$ is the shared representation, $h_0$ and $h_1$ are the potential outcome hypothesis functions, $L(\cdot)$ refers to the squared loss if $Y$ is continuous and cross-entropy if $Y$ is binary, and $\mathcal{R}(\cdot)$ is an L2-regularisation term.

**Remark** (Balanced Representations and Causal Identifiability)**.** *Inspired by ideas from domain adaptation, Shalit et al. (2017) show that learning invertible feature maps $\Phi$ that minimize the distance between treatment groups in feature space leads to minimization of a generalization error bound. Nonetheless, we refrain from using balanced representations here (and hence rely on Shalit et al. (2017)'s TARNet instead of their proposed counterfactual regression method based on balanced representations (CFR)), because minimizing Shalit et al. (2017)'s proposed loss function associated with CFR does not necessarily result in invertible representations (Johansson et al. (2019), Zhang et al. (2020)) – which can be detrimental in the causal inference setting. If information is discarded by artificially balancing treatment groups in a new feature space, this can re-introduce selection bias (which was controlled for in the original feature space!). While we do not investigate this line of thought further here, it would be possible to add an invertibility-penalty similar to the one used in Zhang et al. (2020) to Shalit et al. (2017)'s CFR loss function to circumvent this problem. We also note that representations do not necessarily have to be invertible for causal identification – rather, representations have to preserve all identifying conditional independence relationships. Formally, $\Phi(X)$ has to satisfy $W \perp\!\!\!\perp X | \Phi(X)$ for confounders $X$ which means that it should take the role of a balancing score (Rosenbaum and Rubin, 1983). Hence, it would not be problematic to discard features without confounding effect, even though this leads to non-invertible representations.*

**SNet-2 (DragonNet, adapted from Shi et al. (2019))**

$$\min_{h_0, h_1, \Phi} \frac{1}{n} \sum_{i=1}^{n} [L(h_{W_i}(\Phi(X_i)), Y_i) + CrossEntropy(h_\pi(\Phi(X_i)), W_i)] + \lambda \mathcal{R}(h_0, h_1, h_\pi) \tag{6}$$

where the only difference with SNet-1 arises from the additional hypothesis function $h_\pi$ for the propensity score, which is also learnt using the representation $\Phi$.

**SNet-3 and SNet** Hassanpour and Greiner (2020)'s DR-CFR (SNet-3) is built upon three instead of one representation, where one affects outcome only ($\Phi_O$), one affects treatment only ($\Phi_W$) and one affects both and is hence a true confounder ($\Phi_C$). We use the following adapted loss function to implement SNet-3 and SNet (where SNet uses the same loss function but adds two further representations $\Phi_{\mu_0}$ and $\Phi_{\mu_1}$ which affect only the respective treatment group.)

$$\begin{aligned} \min_{h_0, h_1, \Phi_O, \Phi_C, \Phi_W} \frac{1}{n} \sum_{i=1}^{n} [L(h_{W_i}(\Phi_O(X_i), \Phi_C(X_i)), Y_i) + CrossEntropy(h_\pi(\Phi_C(X_i), \Phi_W), W_i)] + \\ \lambda \mathcal{R}(h_0, h_1, h_\pi) + \gamma \mathcal{R}_O(\Phi_O, \Phi_C, \Phi_W) \end{aligned} \tag{7}$$

As in Wu et al. (2020)'s adaptation of DR-CFR, we also add an orthogonalization term $\mathcal{R}_O$ to our implementation, which ensures that each variable in $X$ affects only one of the 3 (5) representations. This is necessary because – as representations and outcome functions are learned jointly – the learned representations are not identifiable otherwise. To enforce separation and hence specialisation of each representation, we add a regularization term that penalizes whenever a variable enters two representations. Let $W^{1,\Phi_k}$ be the first weight matrix in representation $\Phi_k$ such that $XW^{1,\Phi_k}$ is the first pre-activation in the representation layer. Whether variable $j$ enters representation $\Phi_k$ can be measured by $\bar{W}_{\Phi_k,j} = \sum_u |W^{1,\Phi_k}_{j,u}|$, and the orthogonalization term $\mathcal{R}_O$ simply consists of all cross-products $\bar{W}_{\Phi_k,j} \times \bar{W}_{\Phi_l,j}$ of the different representation contributions. While this does not force hard-decomposition, it does penalize a variable entering multiple representations, leading to good disentanglement in practice.

### 3.3 IMPLEMENTATION DETAILS

In our implementations, we use components similar to those used in Shalit et al. (2017) for all networks. In particular, we use dense layers with exponential linear units (ELU) as nonlinear activation functions. We train with Adam (Kingma and Ba, 2015), minibatches of size 100, and use early stopping based on a 30% validation split. For SNet-1 and SNet-2, the representation $\Phi$ consists of 3 layers with 200 units, while for SNet-3 $\Phi_C$ has 150 units and $\Phi_O$ and $\Phi_W$ have 50. In SNet, $\Phi_C$ and $\Phi_W$ have 100 units, while $\Phi_O$, $\Phi_{\mu_0}$ and $\Phi_{\mu_1}$ have 50 units. For hypothesis functions without shared layers (TNet, the second step regressions, and the propensity score in SNet-1), each hypothesis function gets 3 layers of 200 units of its own. Finally, each hypothesis function (output head) consists of 2 additional layers with 100 units and a final prediction layer (with sigmoid-activation for the propensity score). This set-up ensures that each estimated function ($\hat{\mu}_w(x)$, $\hat{\pi}(x)$ and $\hat{\tau}(x)$) has access to the same amount of layers and units in total, and each architecture can hence represent equally complex nuisance functions. We set $\lambda = 0.0001$ throughout, and $\gamma = 0$ in the IHDP experiments and $\gamma = 0.01$ in the simulation study. All models were implemented using jax (Bradbury et al., 2018). Sklearn-style implementations for all models are available at `https://github.com/AliciaCurth/CATENets` and at `https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/CATENets`.

## 4 EXPERIMENTS

The code used to perform all experiments is available at `https://github.com/AliciaCurth/CATENets` and at `https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/CATENets`.

### 4.1 Simulation set-up

We use a simulation set-up that is partially inspired by that used in Hassanpour and Greiner (2020). In particular, across all settings, we use $d = 25$ multivariate normal covariates $X$ which are simulated in disjoint subsets $X_s$ of $X$, with size $d_s$ according to $X_s \sim \mathcal{N}(0, I)$. For each setting and each training sample size $n \in \{1000, 2000, 5000, 10000\}$ we draw 10 independent training samples of size $n$ and test samples of size 500.

For settings (i) and (ii), we use covariates $X_C$, which are confounders affecting both outcome and treatment assignment, and $X_O$, affecting only outcomes. Both $X_C$ and $X_O$ are composed of 5 covariates. We model the baseline outcome as

$$\mu_0(x) = \mathbb{1}^\top X^2_{CO} \tag{8}$$

where $X_{CO} = [X_C, X_O]$, $\mathbb{1}$ is the unit vector and $X_{CO}$ is squared elementwise.

Treatments are sampled as a Bernoulli random variable using the propensity score:

$$\pi(x) = expit(\xi(\frac{1}{d_c}\mathbb{1}^\top X^2_c - \omega)) \tag{9}$$

where $\xi$ determines the extent of the selection bias. In our experiments we set $\xi = 3$. Further, we adaptively set $\omega = median(\frac{1}{d_c}\mathbb{1}^\top X^2_c)$ in each simulation run to center propensity scores (if we would not do so, the squares in the specification would lead to a much larger treatment group than control group) .

In setting (i), where there is no treatment effect, we set $\mu_{1,(i)}(x) = \mu_0(x)$. In setting (ii), we use 5 additional covariates $X_\tau$ to model a treatment effect:

$$\mu_{1,(ii)}(x) = \mu_0(x) + \mathbb{1}^\top X^2_\tau \tag{10}$$

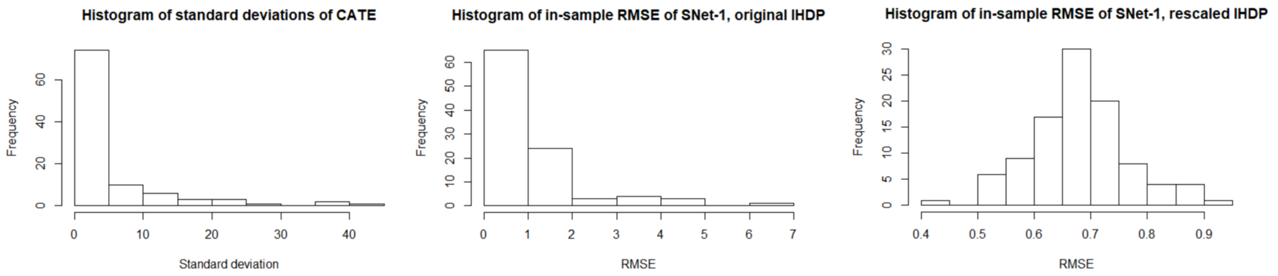

Figure 1: From left to right: Histograms of the standard deviation of CATE, in-sample RMSE of SNet-1 on the original IHDP data-set and the rescaled data-set

For setting (iii), we simulate a setting without confounding ($\pi(x) = 0.5$) where the potential outcome functions are determined by non-overlapping covariate sets, both of dimension 10, i.e.

$$\mu_{0,(iii)} = \mathbb{1}^\top X^2_{\mu_0} \text{ and } \mu_{1,(iii)} = \mathbb{1}^\top X^2_{\mu_1} \tag{11}$$

Finally, in all three settings we compute outcomes as

$$Y_i = W_i \mu_1(X_i) + (1 - W_i)\mu_0(X_i) + \epsilon_i \tag{12}$$

with $\epsilon_i \sim \mathcal{N}(0, 1)$

### 4.2 IHDP data-set

We use an adapted version of the Infant Health and Development Program (IHDP) benchmark used in Shalit et al. (2017) and extensions, created by Hill (2011). The underlying data-set belongs to a real randomized experiment targeting premature infants with low birth weight with an intervention, which contains 25 covariates (6 continuous and 19 binary) capturing aspects related to children and their mothers. The benchmark data-set was created by excluding a non-random proportion of treated individuals, namely those with nonwhite mothers. The final data-set consists of 747 observations (139 treated, 608 control), and overlap is not satisfied (as $\pi(x)$ is not necessarily non-zero for all observations in the control group). While the covariate data is real, the outcomes are simulated according to setting "B" described in Hill (2011), which satisfies $Y(0) \sim \mathcal{N}(exp((X + W)\beta), 1)$ and $Y(1) \sim \mathcal{N}(X\beta - \omega, 1)$ with $W$ an offset matrix and the coefficient $\beta$ has entries in $(0, 0.1, 0.2, 0.3, 0.4)$, where each entry is independently sampled with probabilities (0.6, 0.1, 0.1, 0.1, 0.1). We use the 100 repetitions of the simulation provided by Shalit et al. (2017).

**Rescaled data-set** Rooted in the simulation specification used to obtain the IHDP regression surfaces, we observed that the scale of CATE varied by orders of magnitude across different runs of the simulation, making the RMSE incomparable across runs. We found that by averaging across the data-sets, the relative performance was dominated by runs with high variance in CATE (which are those where many variables have the larger non-zero coefficients), distorting the per-run relative performance we observed. Therefore, we decided to rescale the outcomes of runs where the training set had $\sigma^2_{CATE} > 1$. For these runs, we kept the original error terms $\epsilon_i = Y_i(w) - \mu_{w,i}$ (which were $\mathcal{N}(0, 1)$ across all runs) but rescaled the expected potential outcomes as $\tilde{\mu}_{w,i} = \frac{\mu_{w,i}}{\sigma_{CATE}}$ before adding back the error-term. In Figure 4.2, we plot $\sigma_{CATE}$ as well as the distribution of RMSE in the original and the adapted version of the data-sets for SNet-1, illustrating that only after rescaling the data-set the RMSE results in comparable (and even approximately normal) performance across runs.

### References

Alaa, A. M. and van der Schaar, M. (2018). Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design. In *International Conference on Machine Learning*, pages 129–138.

Bickel, P. J., Ritov, Y., Tsybakov, A. B., et al. (2009). Simultaneous analysis of lasso and dantzig selector. *The Annals of statistics*, 37(4):1705–1732.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Hassanpour, N. and Greiner, R. (2020). Learning disentangled representations for counterfactual regression. In *International Conference on Learning Representations*.

Hill, J. L. (2011). Bayesian Nonparametric Modeling for Causal Inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240.

Johansson, F. D., Sontag, D., and Ranganath, R. (2019). Support and invertibility in domain-invariant representations. *arXiv preprint arXiv:1903.03448*.

Kennedy, E. H. (2020). Optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

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

Raskutti, G., Yu, B., and Wainwright, M. J. (2009). Lower bounds on minimax rates for nonparametric regression with additive sparsity and smoothness. In *Advances in Neural Information Processing Systems*, pages 1563–1570.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Shalit, U., Johansson, F. D., and Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR.

Shi, C., Blei, D., and Veitch, V. (2019). Adapting neural networks for the estimation of treatment effects. In *Advances in Neural Information Processing Systems*, pages 2507–2517.

Stone, C. J. (1980). Optimal rates of convergence for nonparametric estimators. *The annals of Statistics*, pages 1348–1360.

Wu, A., Kuang, K., Yuan, J., Li, B., Zhou, P., Tao, J., Zhu, Q., Zhuang, Y., and Wu, F. (2020). Learning decomposed representation for counterfactual inference. *arXiv preprint arXiv:2006.07040*.

Yang, Y., Tokdar, S. T., et al. (2015). Minimax-optimal nonparametric regression in high dimensions. *The Annals of Statistics*, 43(2):652–674.

Zhang, Y., Bellot, A., and Schaar, M. (2020). Learning overlapping representations for the estimation of individualized treatment effects. In *International Conference on Artificial Intelligence and Statistics*, pages 1005–1014. PMLR.