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

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

Abstract

The need to evaluate treatment effectiveness is ubiquitous in most of empirical science, and interest in flexibly investigating effect heterogeneity is growing rapidly. To do so, a multitude of model-agnostic, nonparametric meta-learners have been proposed in recent years. Such learners decompose the treatment effect estimation problem into separate sub-problems, each solvable using standard supervised learning methods. Choosing between different meta-learners in a data-driven manner is difficult, as it requires access to counterfactual information. Therefore, with the ultimate goal of building better understanding of the conditions under which some learners can be expected to perform better than others *a priori*, we theoretically analyze four broad meta-learning strategies which rely on plug-in estimation and pseudo-outcome regression. We highlight how this theoretical reasoning can be used to guide principled algorithm design and translate our analyses into practice by considering a variety of neural network architectures as base-learners for the discussed meta-learning strategies. In a simulation study, we showcase the relative strengths of the learners under different data-generating processes.

1 INTRODUCTION

Many empirical scientists ultimately aim to assess the causal effects of interventions, policies and treatments by analyzing experimental or observational data using tools from applied statistics. Due to the impressive performance of machine learning (ML) methods on prediction tasks, recent years have seen exciting developments incorporating ML into the estimation of average treatment effects (van der Laan and Rose, 2011; Chernozhukov et al., 2017). While average treatment effects (ATEs) have been the main estimand of interest thus far, data-adaptive, ML-based estimators have even more potential to shape our ability to flexibly investigate heterogeneity of effects across populations. As interest moves towards personalized policy- and treatment design in fields such as econometrics and medicine, the need to accurately estimate the full conditional average treatment effect (CATE) function becomes ubiquitous.

The causal inference communities across disciplines have produced a rapidly growing number of algorithms for CATE estimation in recent years (see e.g. Bica et al. (2020) for an overview). In practice, this leads to the need to select the best model – which is notoriously difficult in treatment effect studies because the ground truth is unobserved. While recent literature has presented promising solutions using data-driven strategies (Rolling and Yang, 2014; Alaa and Van Der Schaar, 2019), we believe that it is equally important to reduce the complexity of the selection task a priori by building greater systematic understanding of the strengths and weaknesses of different algorithms from a theoretical viewpoint.

Here, we put our focus on comparing different so-called meta-learners for binary treatment effect estimation, which are model-agnostic algorithms that decompose the task of estimating CATE into multiple sub-problems, each solvable using *any* supervised learning/ regression method (Künzel et al., 2019). In the theoretical part of this paper (Sections 3 and 4), we consider estimation within a generic nonparametric regression framework, i.e. we assume no known parametric structure, and derive theoretical arguments why one learner may outperform others. In the more practical part (Sections 5 and 6), we compare the em-

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

pirical performance of the different learners using *the same* underlying machine learning method, and consider a variety of neural network (NN) architectures for CATE estimation. Throughout, instead of arguing that one learning algorithm is superior to all others, we aim to highlight how expert knowledge on the underlying data-generating process (DGP) can narrow the choice of algorithms a priori and guide model design.

Contributions Our contributions are three-fold: First, we provide theoretical insights into nonparametric CATE estimation using meta-learners. We propose a new classification of meta-learners inspired by the ATE estimator taxonomy, categorizing algorithms into four broad classes: one-step plug-in learners and three types of two-step learners, which use unbiased pseudo-outcomes based on regression adjustment (RA), propensity weighting (PW) or both (DR), as illustrated in Figure 1. We present an analysis of the theoretical properties of the learners and discuss resulting theoretical criteria for choosing between them. While both plug-in and DR-learner have been previously analyzed, our analysis and discussion of RA- and PW-learner are – to the best of our knowledge – new. Second, we compare four existing model architectures for CATE estimation using NNs and propose a new architecture which generalizes existing approaches. These architectures allow for different degrees of information sharing between nuisance parameter estimators, and we highlight the (dis-)advantages of different architectures. We also provide a suite of sklearn-style implementations for all architectures and meta-learners we consider[1].
Third, we illustrate our theoretical arguments in simulation experiments, demonstrating how differences in DGPs and sample size influence the relative performance of different learners empirically. By considering how to best combine model architectures and meta-learner strategies, we also attempt to bridge the gap between the relatively disjoint literatures on meta-learners and end-to-end CATE estimation using NNs.

1.1 Related Work

We restrict our attention to so-called 'meta-learners' for CATE – model-agnostic algorithms that can be implemented using *any arbitrary* ML method. Künzel et al. (2019) appear to be the first paper explicitly discussing meta-learning strategies for CATE estimation, of which they consider and named three in detail: The S-learner (*single* learner), in which the treatment indicator is simply included as an additional feature in otherwise standard regression, the T-learner (*two* learners) which fits separate regression functions for each treatment group and then takes differences, and the X-learner, a two-step regression estimator that uses each observation twice (see discussion in Section 3). Nie and Wager (2020) propose a two-step algorithm that estimates CATE using orthogonalization with respect to the nuisance functions, which they dub R-learner as it is based on Robinson (1988). Finally, Kennedy (2020) proposes the DR-learner (*doubly robust* learner), a two-step algorithm that uses the expression for the doubly robust augmented inverse propensity weighted (AIPW) estimator (Robins and Rotnitzky, 1995) as a pseudo-outcome in a two-step regression set-up. With the exception of the DR-learner, the current naming strategy of meta-learners is surprisingly disjoint from the naming of ATE estimators, resulting in names that do not necessarily reflect the statistical concepts the learners are based on. To build better intuition and to facilitate principled theoretical analyses, we re-categorize meta-learning strategies in Section 3.

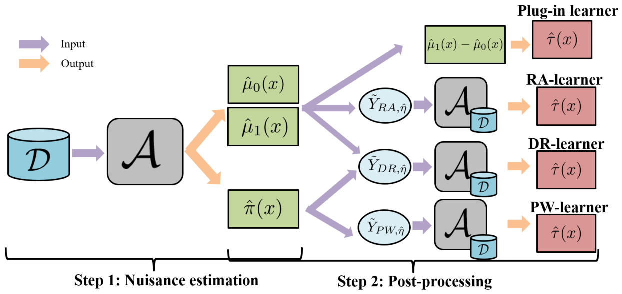

Figure 1: High-level overview of the four meta-learning strategies considered in this paper. $\mathcal{A}$ refers to a generic regression algorithm and $\mathcal{D}$ refers to input data, both of which are used to estimate the propensity score $\hat{\pi}(x)$, the potential outcomes $\hat{\mu}_w(x)$ and CATE $\hat{\tau}(x)$, and to compute the pseudo-outcomes $\tilde{Y}_{l,\hat{\eta}}$.

Our theoretical considerations build on the statistical analyses of the fundamental limits of CATE estimation presented for Bayesian nonparametrics in Alaa and van der Schaar (2018a,b), as well as the analyses of frequentist estimation using S-, T- and X-learner in Künzel et al. (2019) and the DR-learner in Kennedy (2020). All prior work considers only a subset of (variations of) the meta-learners we consider here, and can hence not lead to comprehensive practical advice on choosing between all strategies.

While our theoretical analyses of meta-learners apply to generic estimators, in the practical part of this paper we focus on instantiations using standard feed-forward neural networks (NNs) and NN-based repre-

[1] The code is available at `https://github.com/AliciaCurth/CATENets` and at `https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/CATENets`

sentation learning. This choice is motivated by interesting recent implementations (Johansson et al., 2016; Shalit et al., 2017; Hassanpour and Greiner, 2020; Shi et al., 2019) of hybrids of Künzel et al. (2019)'s S- and T-learners, which build on ideas from representation learning (Bengio et al., 2013) and multi-task learning (Caruana, 1997), sharing some information between regression tasks. We discuss these in detail in Section 5. As the meta-learners can be used with arbitrary regression estimators, a wide range of other ML methods have been used in practice, and popular choices include Bayesian Additive Regression Trees (Hill, 2011) and random forests (Künzel et al., 2019). Note that, in contrast to the model-agnostic nature of the meta-learners, there also exist a variety of CATE estimators that rely on a *specific ML method* – e.g. generative adversarial networks (Yoon et al., 2018), deep kernel learning (Zhang et al., 2020) or generalized random forests (Athey et al., 2019) – which therefore fall outside the scope of this paper.

2 PROBLEM DEFINITION

Assume we observe a sample $\mathcal{D} = \{(Y_i, X_i, W_i)\}_{i=1}^n$, with $(Y_i, X_i, W_i) \overset{i.i.d.}{\sim} \mathbb{P}$. Here, $Y \in \mathcal{Y}$ is a continuous or binary outcome of interest, $X_i \in \mathcal{X} \subset \mathbb{R}^d$ a d-dimensional covariate vector of possible confounders and $W_i \in \{0, 1\}$ is a binary treatment, which is assigned according to propensity score $\pi(x) = \mathbb{P}(W = 1|X = x)$, with marginal treatment assignment probability $p_\pi = \mathbb{P}(W = 1)$. Using the Neyman-Rubin potential outcomes framework (Rubin, 2005), our main interest lies in the individualized treatment effect: the difference between the potential outcomes $Y_i(0)$ if individual i does not receive treatment ($W_i = 0$) and $Y_i(1)$ if treatment is administered ($W_i = 1$). However, by the *fundamental problem of causal inference*, only one of the potential outcomes is observed, since $Y_i = W_i Y_i(1) + (1 - W_i)Y_i(0)$. Therefore, in line with the majority of existing literature, we focus on estimating the conditional average treatment effect (CATE),

$$\tau(x) = \mathbb{E}_{\mathbb{P}}[Y(1) - Y(0)|X = x] \tag{1}$$

the expected treatment effect for an individual with covariate values $X = x$.

It is well known that the identification of causal effects from observational data hinges on the imposition of untestable assumptions. Here, we consider estimation under the standard assumptions:

Assumption 1. *[Consistency, unconfoundedness and overlap] Consistency: If individual i is assigned treatment w_i, we observe the associated potential outcome $Y_i = Y_i(w_i)$. Unconfoundedness: there are no unobserved confounders, such that $Y(0), Y(1) \perp\!\!\!\perp W|X$. Overlap: treatment assignment is non-deterministic, i.e. $0 < \pi(x) < 1$, $\forall x \in \mathcal{X}$.*

Under assumption 1, CATE can be written as $\tau(x) = \mu_1(x) - \mu_0(x)$ for $\mu_w(x) = \mathbb{E}_{\mathbb{P}}[Y|W = w, X = x]$, and can be estimated from observational data using the meta-learners we discuss. We assume no known parametric form for $\tau(x)$, any of the nuisance parameters $\eta = (\mu_0(x), \mu_1(x), \pi(x))$ or error distributions in $\mathbb{P}$, leaving us with a *nonparametric* estimation problem. Throughout the theoretical analysis we consider generic nonparametric regression estimators, and use NNs in the experiments, yet all strategies could be used directly with other methods, e.g. random forests.

3 CATEGORIZING CATE META-LEARNERS

To improve our ability to analyze the meta-learner's theoretical performance in a structured manner in the following section, we propose a high-level classification of CATE meta-learners that follows the well-known ATE taxonomy of estimators (see e.g. Imbens (2004)). We prefer this naming strategy because it builds on existing intuition and classifies learners by the characteristics that reflect their statistical properties. Therefore, we suggest to divide meta-learners into *one-step plug-in learners* – learners that output two regression functions which can then be differenced – and *two-step learners*, based on a *regression adjustment (RA)*, *propensity weighting (PW)* or *doubly robust (DR)* strategy, outputting a CATE function directly. These strategies are illustrated in Figure 1.

We define one-step plug-in learners as those who obtain regression functions $\hat{\mu}_w(x)$ from the observed data, and estimate CATE directly as $\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x)$. This is the strategy underlying Künzel et al. (2019)'s S- and T-learners. We do consider it important to keep a distinction between S- and T-learner as special cases of plug-in learners, because they differ in an important dimension: namely the amount of information shared between the potential outcome estimators. We use these terms to summarize somewhat broader estimator classes than Künzel et al. (2019): We consider as T-learners those that separate estimation of the nuisance functions into disjoint sub-tasks (of which are *t*wo or *t*hree, depending on estimation of $\pi(x)$) while S-learners *s*hare some information between nuisance estimators, which encompasses more strategies than just including the treatment indicator as a covariate (e.g. joint feature selection, or the strategies for NNs we discuss in Section 5).

We define two-step learners by their approach of using a first stage to obtain plug-in estimates $\hat{\eta}$ of (a subset of) the nuisance parameters $\eta = (\mu_0(x), \mu_1(x), \pi(x))$, and then a second stage to obtain an estimate $\hat{\tau}(x)$ by regressing a pseudo-outcome $\tilde{Y}_{\hat{\eta}}$ (based on nuisance estimates $\hat{\eta}$) on X directly. To do so, we consider pseudo-outcomes for which it holds that $\mathbb{E}_{\mathbb{P}}[\tilde{Y}_{\eta}|X = x] = \tau(x)$ (i.e. they are unbiased for CATE when η is known), for which there are three straightforward strategies inspired by ATE estimators:

(1) We propose an RA-learner, which uses the regression-adjusted pseudo-outcome

$$\tilde{Y}_{RA,\hat{\eta}} = W(Y - \hat{\mu}_0(X)) + (1 - W)(\hat{\mu}_1(X) - Y) \quad (2)$$

in the second step. Künzel et al. (2019)'s X-learner is a variant of this estimator: Instead of performing one regression in the second step, they suggest performing two separate regressions for each term in the sum, leading to two CATE estimators $\hat{\tau}_1(x)$ and $\hat{\tau}_0(x)$ that should then be combined into a final estimate using $\hat{\tau}(x) = g(x)\hat{\tau}_0(x) + (1 - g(x))\hat{\tau}_1(x)$ for some weighting function $g(x)$. For $g(x) = 1 - \pi(x)$, the two estimators coincide in expectation. We prefer the RA-learning strategy here, as it does not require choice of 'hyperparameter' $g(x)$.

(2) Inspired by inverse propensity weighted (IPW) estimators, we consider a PW-learner with associated pseudo-outcome based on the Horvitz-Thompson transformation (Horvitz and Thompson, 1952)

$$\tilde{Y}_{PW,\hat{\eta}} = \left(\frac{W}{\hat{\pi}(X)} - \frac{1 - W}{1 - \hat{\pi}(X)}\right) Y \quad (3)$$

(3) Finally, the DR-learner (Kennedy, 2020) has pseudo-outcome

$$\begin{aligned}\tilde{Y}_{DR,\hat{\eta}} = \left(\frac{W}{\hat{\pi}(X)} - \frac{(1 - W)}{1 - \hat{\pi}(X)}\right) Y + \\ \left[\left(1 - \frac{W}{\hat{\pi}(X)}\right)\hat{\mu}_1(x) - \left(1 - \frac{1 - W}{1 - \hat{\pi}(X)}\right)\hat{\mu}_0(X)\right]\end{aligned} \quad (4)$$

which is based on the doubly-robust AIPW estimator (Robins and Rotnitzky, 1995) and is hence unbiased if either propensity score *or* outcome regressions are correctly specified. Variants of the DR-learner have previously also been studied in e.g. Lee et al. (2017); Fan et al. (2020).

The R-learner proposed in Nie and Wager (2020) does not fall in any of these categories. While all learners considered above are inherently model-agnostic, i.e. can be implemented using any off-the-shelf ML method, the R-learner requires specific model-fitting procedures (e.g. the ability to manipulate a loss function). We therefore do not consider this estimator further here, also because an in-depth theoretical analysis is given in Nie and Wager (2020).

4 THEORETICAL ANALYSES OF CATE META-LEARNERS

In this section we consider the theoretical behavior of the four different types of CATE meta-learners. While the asymptotic properties of both plug-in and DR-learner have been previously analyzed, our analyses of both RA- and PW-learner, as well as the comparison between all four strategies in asymptotic and finite sample settings, are new and are meant to provide insights to guide principled choice between algorithms.

Throughout, we denote by $\tau(x)$ the true CATE and by $\hat{\tau}_{l,\hat{\eta}}(x)$ the output of learner l. For two-step estimators we denote by $\hat{\tau}_{l,\eta}(x)$ the output of the second stage regression if we had oracle-access to the nuisance parameters. Further, for an estimator $\hat{f}(x)$ of $f(x)$, we denote by $\epsilon_{sq}(\hat{f}(x)) = \mathbb{E}[(\hat{f}(x) - f(x))^2]$ its expected squared error. Finally, we use $a \lesssim b$ to indicate $a \leq Cb$ for some universal constant C. For the purpose of the theoretical analysis in this section, we make the following two assumptions:

Assumption 2. *[Assumptions on estimators] We assume for the propensity score estimates that $\delta \leq \hat{\pi}(x) \leq 1 - \delta$ for $\delta > 0$. Further, all regression estimators fulfill the mild assumptions characterized in Kennedy (2020)'s Theorem 1 (see supplement).*

Assumption 3. *[Assumptions on true DGP] We assume that $\mu_w(x)$ is α_w-smooth, $\pi(x)$ is β-smooth and $\tau(x)$ is γ-smooth, and all are estimable at Stone (1980)'s minimax rate of $n^{\frac{-p}{2p+d}}$ for a p-smooth function (which has p continuous and bounded derivatives). Further, the potential outcomes and the propensity scores are bounded, i.e. $|\mu_w(x)| \leq C$ and $\omega \leq \pi(x) \leq 1 - \omega$ for $C, \omega > 0$.*

Asymptotic Behavior First, we consider the (asymptotic) behavior of $\epsilon_{sq}(\hat{\tau}_{l,\hat{\eta}}(x)) = \mathbb{E}[(\hat{\tau}_{l,\hat{\eta}}(x) - \tau(x))^2]$. For a generic one-step plug-in estimator $\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x)$ we almost trivially have that

$$\epsilon_{sq}(\hat{\tau}(x)) \leq 2[\epsilon_{sq}(\hat{\mu}_1(x)) + \epsilon_{sq}(\hat{\mu}_0(x))] \lesssim n^{\frac{-2\alpha_0}{2\alpha_0+d}} + n^{\frac{-2\alpha_1}{2\alpha_1+d}}$$

by $(a + b)^2 \leq 2(a^2 + b^2)$ and assumption 3. The second inequality holds asymptotically for plug-in estimators based on T-strategies, but not necessarily for the S-learners because information-sharing might restrict the set of possible potential outcome functions.

The three other learners all require two regression stages and are hence more difficult to analyze. However, under the assumption that we perform the two regression stages on two separate samples $\mathcal{D}_0$ and $\mathcal{D}_1$ of size n, we can apply Kennedy (2020)'s Theorem 1 , yielding that $\epsilon_{sq}(\hat{\tau}_{l,\hat{\eta}}(x))$

$$\lesssim \underbrace{\epsilon_{sq}(\hat{\tau}_{l,\eta}(x))}_{\lesssim n^{\frac{-2\gamma}{2\gamma+d}}} + \underbrace{\mathbb{E}[(\mathbb{E}[\tilde{Y}_{\hat{\eta}}(x)|X=x,\mathcal{D}_0]-\tau(x))^2]}_{=R^2_{l,\hat{\eta}}(x)}$$

for all three learners l. As the oracle term is of fixed order across all estimators, the remainder term $R^2_{l,\hat{\eta}}(x)$, which we analyze in the following theorem (proof and additional analyses in supplement), will determine relative asymptotic performance.

Theorem 1. *[Learner-specific remainders] Under assumptions 2 and 3 and using two-step estimation on two separate samples $\mathcal{D}_0$ and $\mathcal{D}_1$ of size n, we have that*
(1) for the RA-learner:

$$R^2_{RA,\hat{\eta}}(x) \leq 2(1-\omega)^2 \left(\sum_w \epsilon_{sq}(\hat{\mu}_w(x))\right) \lesssim n^{\frac{-2\alpha_0}{2\alpha_0+d}} + n^{\frac{-2\alpha_1}{2\alpha_1+d}}$$

(2) for the PW-learner:

$$R^2_{PW,\hat{\eta}}(x) \leq \frac{4C^2}{\delta^2}\epsilon_{sq}(\hat{\pi}(x)) \lesssim n^{\frac{-2\beta}{2\beta+d}}$$

(3) (Kennedy (2020), Theorem 2 and Corollary 1) If $\pi(x)$ and $\mu_w(x)$ are fit on separate sub-samples, we have for the DR-learner that:

$$R^2_{DR,\hat{\eta}}(x) \leq \frac{2}{\delta^2}\epsilon_{sq}(\hat{\pi}(x)) \left(\sum_w \epsilon_{sq}(\hat{\mu}_w(x))\right) \lesssim n^{-2(\min_w \frac{\alpha_w}{2\alpha_w+d} + \frac{\beta}{2\beta+d})}$$

By asymptotic properties, we thus prefer the DR-learner over the other two-step learners, and the PW-learner over the RA-learner if $\beta > min_w \alpha_w$, i.e. if the propensity score is easier to estimate than the outcome regressions. Further, if $\min_w \frac{\alpha_w}{2\alpha_w+d} + \frac{\beta}{2\beta+d} > \frac{\gamma}{2\gamma+d}$, the DR-learner attains the oracle rate, and so do PW- and RA-learner if $\beta > \gamma$ and $min_w \alpha_w > \gamma$, respectively. The latter, however, is unlikely since it is commonly assumed that in practice $\tau(x)$ is simpler than $\mu_0(x)$ (Künzel et al., 2019). Asymptotically, we therefore expect RA- and plug-in learner to perform similarly. In the case where $\tau(x)$ is of similar complexity as the potential outcomes functions, plug-in and two-step learners can be expected to perform similarly. Instead of relying on assumed smoothness of the nuisance functions, similar analyses can be performed by relying on different assumptions on the problem structure. In the supplement, we consider sparsity instead of smoothness, which leads to analogous conclusions in terms of relative performance of the different learners.

While in reality it is unlikely to have exact knowledge on the theoretical properties of the nuisance functions, the considerations above are also useful when an expert can give insight on the *relative* complexity of the underlying functions. Further, in experimental settings, when propensity scores are *known*, the remainder terms of PW- and DR-learner will be *exactly* zero, rendering their performance equal to the oracle rate.

Considerations for Finite Samples However, as we will discuss next, in smaller sample regimes, we cannot rely on convergence rates only. A first reason for this derives from the fact that while the oracle term $\mathbb{E}[(\hat{\tau}_{l,\eta}(x)-\tau(x))^2]$ is of the same order for all estimators, the variance of inverse propensity weighted estimators is well-known to be high, particularly when propensity scores are extreme. Additionally, the pseudo-outcome associated with the PW-learner has a very high variance even when propensity scores are constant and known, because it then still lies in $\{-2Y(0), 2Y(1)\}$. This – as we will demonstrate in the experiments – can lead to very poor empirical performance of the PW-learner. Similarly, for δ small, the constants in the error bounds of PW- and DR-learner in Theorem 1 can be large, possibly leading to relatively better performance of the RA-learner in smaller samples.

Second, so far we did not consider *selection bias* – $\pi(x) \neq 0.5$ for some x – which matters in finite sample regimes, as it results in each regression surface not being fit optimally. Consider the expected (integrated) error on the treated outcome surface, for which it is straightforward to show (see supplement) that:

$$\mathbb{E}_{X\sim\mathbb{P}(\cdot)}[(\hat{\mu}_1(X)-\mu_1(X))^2] = \mathbb{E}_{X\sim\mathbb{P}(\cdot|W=1)}[w(X)(\hat{\mu}_1(X)-\mu_1(X))^2] \quad (5)$$

where $w(X) = p_\pi(1+\frac{1-\pi(X)}{\pi(X)})$. When $\hat{\mu}_1(x)$ is fit on factual data only, this corresponds to setting $w(x)=1$ for all x, which gives too much weight to samples with large $\pi(x)$. Combining this with assumption 3, we have for one-step learners that $\mathbb{E}_{X\sim\mathbb{P}(\cdot)}[(\hat{\tau}(X)-\tau(X))^2]$

$$\leq \frac{2}{\omega}\big(\mathbb{E}_{X\sim\mathbb{P}(\cdot|W=1)}[(\hat{\mu}_1(X)-\mu_1(X))^2] + \mathbb{E}_{X\sim\mathbb{P}(\cdot|W=0)}(\hat{\mu}_0(X)-\mu_0(X))^2]\big)$$

which highlights why selection bias does not matter asymptotically, but can be severe in finite samples if ω is small (a similar conclusion was reached in Alaa and van der Schaar (2018b)). It also shows that the degree of overlap, which determines ω, is of paramount importance. Because the second stage regressions use all data to estimate $\tau(x)$ directly, and hence consider $X \sim \mathbb{P}(\cdot)$, this can correct for sub-optimal weighting in the first stage.

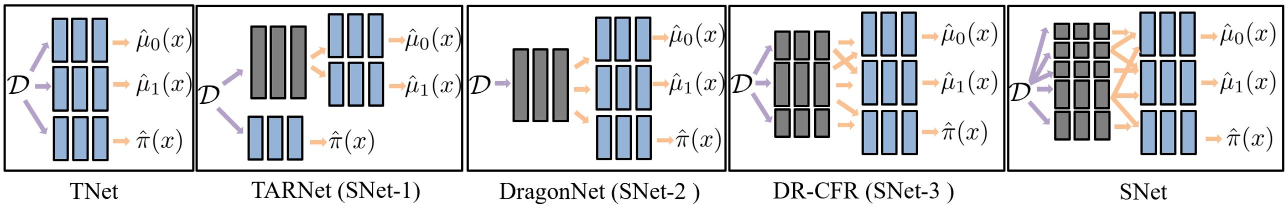

Figure 2: Overview of five possible model architectures for one-step estimation of nuisance parameters, involving different levels of sharing information between tasks (representation layers in gray, task-specific layers in blue)

Overall, we conclude that a second stage regression can act like a 'regularizer' on the first stage output, removing some of the bias induced by regularization and overfitting in finite samples. We note that the DR-learner does so optimally from a theoretical viewpoint, as it can also be shown to build on the notion of nonparametric plug-in bias removal via influence functions (Curth et al., 2020). Nonetheless, we note that two-step learners generally need to estimate more parameters on the same amount of data (or, if sample splitting is used, on less), which could in practice also lead to higher variance and worse performance in low sample regimes.

5 CATE ESTIMATION USING NEURAL NETWORKS

In the practical part of this paper, we use feed-forward NNs as nuisance estimators for each meta-learner. The simplest NN-based implementation of each learner consists of using a separate network for each regression task (a TNet), which would be a good choice asymptotically, allowing for arbitrarily different regression surfaces.

However, as we alluded to in Section 3, it can be useful to share information between nuisance estimation tasks in finite samples. That is, it may be more *efficient* to share data between the two regression tasks if $\mu_1(x)$ and $\mu_0(x)$ are similar, which would be the case if they were supported on similar covariates and $\tau(x)$ is not too complex. Therefore, next to a simple T-learner (TNet), we consider a class of model architectures we refer to as SNets because they are based on *s*haring information between nuisance estimation tasks using representation learning. The resulting one-step architectures for nuisance estimation are visualized in Fig. 2, and we discuss implementation details and loss-functions in the supplement.

Existing SNet Architectures Building on the success of representation learning on a variety of learning tasks (Bengio et al., 2013), Shalit et al. (2017) introduce the idea of learning a shared input representation for the two potential outcome regressions. Formally, this entails jointly learning a map $\Phi : \mathcal{X} \rightarrow \mathcal{R}$, representing *all* data in a new space, and two regression heads $\mu_w : \mathcal{R} \rightarrow \mathcal{Y}$, fit using only the data of the corresponding treatment group. This results in TARNet (Shalit et al., 2017), which we will also refer to as SNet-1 because it results in the simplest way of sharing information between tasks. Shi et al. (2019)'s DragonNet (SNet-2) takes this idea one step further and learns a representation space from which both $\pi(x)$ and $\mu_w(x)$ can be learned, ensuring that confounders are sufficiently controlled for. While Shi et al. (2019) used this architecture, combined with ideas from the targeted maximum likelihood estimation framework, to estimate *average* treatment effects, we use only their model architecture for CATE estimation. Finally, Hassanpour and Greiner (2020)'s DR-CFR (SNet-3) learns three representations which are used to model *either* propensity score, potential outcome regressions or both, respectively[2].

Underlying Assumptions and a New Architecture Each existing architecture reflects implicit assumptions on the underlying structure of the CATE estimation problem. Therefore, explicitly characterizing assumptions can help choosing between architectures in practice, and guide improvements by identifying shortcomings. SNet-1 builds on the assumption that there exists a common feature space underlying both $\{\mu_w(x)\}_{w\in\{0,1\}}$, while SNet-2 assumes that $\pi(x)$ can *also* be represented in the same space. SNet-3 is built on the assumption that there exist three separate sets of features, determining $\mu_w(x)$ and/or $\pi(x)$.

Reflecting on these assumptions, we note that there is one important case missing: We wish to explicitly allow for the existence of features that affect *only one* of the potential outcome functions. This is driven by medical applications where one distinguishes between markers that are prognostic (of outcome) regardless of treatment status or predictive (of treatment effectiveness) (Ballman, 2015). Therefore, we propose a

[2] To limit the number of possible models, we consider only straightforward plug-in strategies, and do not use the propensity heads for re-weighting within the loss function (as Hassanpour and Greiner (2020) do).

final model architecture that learns five representations[3], also allowing potential-outcome regressions to depend on only a subset of shared features. We will simply refer to this architecture as SNet because it *encompasses* all existing SNet-architectures and even the TNet as special cases. That is, if we increase the width of the $\mu_w(x)$-specific representations while reducing the shared representations, SNet will approach TNet. If we do the reverse, the proposed architecture approaches SNet-3, which can in turn become either SNet-1 or SNet-2 by changing what is shared between $\mu_w(x)$ and $\pi(x)$. We expect that such a general architecture should perform best *on average* in absence of knowledge on the underlying problem structure, but may be more difficult to fit than simpler models.

Further Practical Considerations Two-step learners are most commonly implemented using independently trained "vanilla" nuisance estimators, yet, as we investigate in the experiments, all SNet architectures could also be used to estimate nuisance parameters in the first step. Further, while our theoretical analyses rely on sample splitting for two-step learners, we observed that using all data for both steps can work better in practice (particularly in small samples). If estimation with theoretical guarantees is desired yet data is scarce, it can be useful to rely not on sample splitting, but on *cross-fitting* (Chernozhukov et al., 2018) to obtain valid nuisance function estimates for all observations in the sample, which can then all be used for a second stage regression. Which strategy to use involves a trade-off between precision in estimation and computational complexity. While we implement all strategies in our code base, we do not use any form of sample splitting in our experiments. Finally, a convenient by-product of splitting the causal inference task into a series of multiple supervised learning tasks is that hyper-parameters can be tuned using factual hold-out data only – which can be done in both regression stages, if data is split appropriately.

6 EXPERIMENTS

In this section we supplement our theoretical analyses with experimental evidence to demonstrate the empirical performance of all learners under different DGPs, using both fully synthetic data and the well-known semi-synthetic IHDP benchmark. In addition to verifying the theoretical properties of the different learners empirically, we consider it of particular interest to gain insight into how to best use the different NN architectures as nuisance estimators for the two-step learners.

Throughout the experiments, we fixed equivalent hyper-parameters across all model architectures (based on those used in Shalit et al. (2017)), ensuring that every estimator (output head) has access to the same total amount of hidden layers and units, and effectively used each learner 'off-the-shelf'. To ensure fair comparison across learners, we implemented every architecture in our own python code base[4]; for implementation details, refer to the supplement. Throughout, we consider performance in terms of the Root Mean Squared Error (RMSE) of estimation of $\tau(x)$, also sometimes referred to as the precision of estimating heterogeneous effects (PEHE) criterion (Hill, 2011).

6.1 Synthetic Experiments

We simulate data to investigate the relative performance of the different learners across different underlying DGPs and sample sizes. Throughout, we consider $d = 25$ multivariate normal covariates, of which we let subsets determine $\mu_w(x)$ and $\pi(x)$. To highlight scenarios under which different learners can be expected to perform well, we consider three (highly stylized) settings: (i) $\tau(x) = 0$, and $\mu_0(x)$ depends on 5 covariates influencing outcome and 5 confounders (which influence also $\pi(x)$), (ii) the same set-up as (i), but with $\tau(x)$ nonzero and supported on 5 additional covariates and (iii) $\mu_1(x)$ and $\mu_0(x)$ depend on disjoint covariate sets, making $\tau(x)$ the most difficult function to estimate (no confounders). All DGPs are discussed in detail in the supplement. In all simulations, we evaluate performance on 500 independently generated test-observations, and average across 10 runs.

When comparing the 5 plug-in architectures (Fig. 3), we note that S-architectures always improve upon the TNet in small samples when $\{\mu_w\}_{w \in \{0,1\}}$ share some structure. Even when $\mu_1(x)$ and $\mu_0(x)$ are very different, shared layers can help by filtering out noise covariates. Further, the flexibility of the respective SNet architecture seems to indeed drive performance. As expected, the general SNet performs best on average, and is the only architecture to outperform TNet when $\mu_1(x)$ and $\mu_0(x)$ are very different. SNet performs well also in the absence of any treatment effect, which we attribute to the architecture's ability to learn that there are no predictive features to represent using the $\mu_w(x)$-specific representations. Further, as expected,

[3] Note that, when multiple representations and outcome functions are learned jointly, the distinction between representations is not necessarily well-identified. Therefore, for both SNet-3 and SNet we use a regularization term inspired by Wu et al. (2020) that enforces orthogonalization of inputs to the different representation layers, as we discuss in the supplement.

[4] All code is available at `https://github.com/AliciaCurth/CATENets` and at `https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/CATENets`

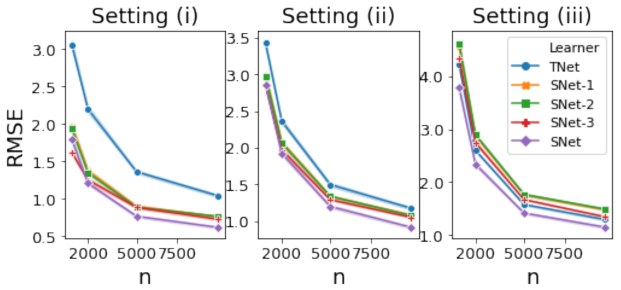

Figure 3: RMSE of plug-in architectures by sample size and across different DGPs. Shaded area indicates one standard error.

the strength of SNet relative to all variants becomes most apparent in relatively larger sample sizes.

When comparing the four learning strategies (Fig. 4), all based on TNet in the first step, we observe that the DR-learner substantially outperforms the others when there is confounding but no treatment effect (setting (i)). Conversely, when CATE is *more* complex than either potential outcome function (setting (iii)) – making the task of learning CATE directly more difficult than learning the potential outcome functions separately – the plug-in learner performs best, as expected. The RA-learner performed best when there is both confounding and a non-trivial treatment effect (setting (ii)). The PW-learner performed poorly in general, which is caused by the very low signal-to-noise ratio and high variance in the associated pseudo-outcome. While all other learning strategies performed very well using equivalent architectures, optimizing the PW-learner would require much stronger regularization and less flexibility (smaller networks) than what we considered here.

To gain further insight to the relative performance of different learning strategies, we interpolate between settings (i) and (ii) by gradually increasing the number of predictive features (features determining $\tau(x)$) at $n = 2000$ in Fig. 5. We observe that the performance of the different S-architectures degrades relative to TNet as the number of predictive features increases, which is to be expected as $\mu_0(x)$ and $\mu_1(x)$ become less similar. We also observe that while the performance gap between TNet and RA-learner remains virtually constant, the DR-learner loses its advantage as CATE becomes less sparse.

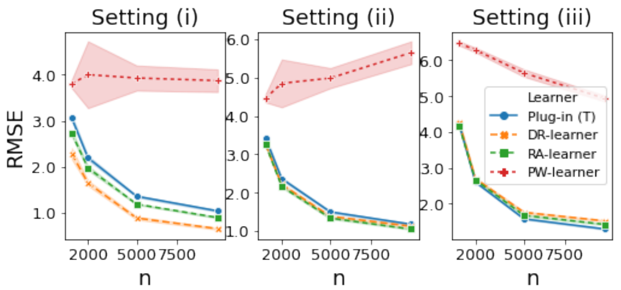

Figure 4: RMSE of different meta-learners by sample size and across different DGPs. Shaded area indicates one standard error.

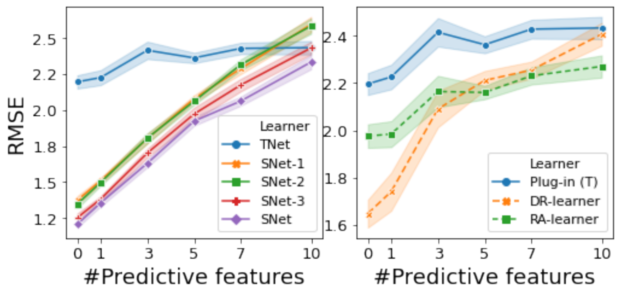

Figure 5: RMSE of different plug-in architectures (left) and meta-learners (right) by number of predictive features at $n = 2000$. Shaded area indicates one standard error.

In Fig. 6 we reconsider setting (i) but with *imbalance* in addition to confounding, by gradually changing the proportion of treated individuals. Comparing the different plug-in architectures, we observe that information sharing has a much larger added value when samples are highly imbalanced. Comparing the different learners, we observe that the performance gap between T- and DR-learner is not impacted by the proportion of treated individuals, but that the RA-learner outperforms the T-learner only for moderate to no imbalance.

Finally, we consider how to best combine nuisance estimators and two-step learners (Fig. 7). We reconsider settings (i) and (ii) where DR- and RA-learner

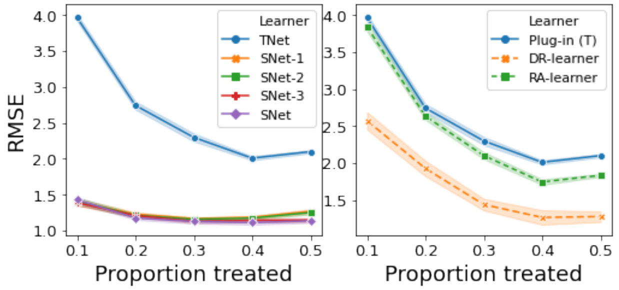

Figure 6: RMSE of different plug-in architectures (left) and meta-learners (right) by proportion of treated individuals in setting (i) at $n = 2000$. Shaded area indicates one standard error.

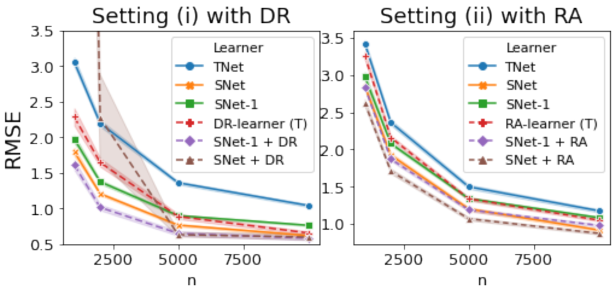

Figure 7: RMSE of different learner-architecture combinations by sample size and across different DGPs. Shaded area indicates one standard error.

performed best, and use SNet-1, which does not estimate $\pi(x)$, and the more general SNet, which does, as nuisance estimators. For the RA-learner, we observe that using SNet, which has better performance on its own, leads to the best RA-learner and conclude that in practice the RA-learner should be combined with an architecture that is expected to best capture the underlying DGP. For the DR-learner, we note that strong dependence between $\hat{\pi}(x)$ and $\hat{\mu}_w(x)$ leads to a slower decaying remainder term theoretically, manifesting in poor empirical performance of using SNet with the DR-learner relative to SNet-1 in smaller samples. Hence, the DR-learner is best combined with a nuisance estimator that does not share information between estimation of $\pi(x)$ and $\mu_w(x)$.

6.2 Semi-synthetic Benchmark: IHDP

Additionally, we deploy all algorithms on the well-known IHDP benchmark, based on real covariates with simulated outcomes. We use an adapted[5] version of the 100 realizations provided by Shalit et al. (2017). The data-set is small ($n = 747$, of which 90% are used in training), imbalanced (19% treated), and there is only partial overlap (Hill, 2011). While the two potential outcome functions are supported on the same covariates, their functional forms are different, making their difference – CATE – the most difficult function to estimate. A more detailed description of the data-set can be found in the supplement.

Table 1: RMSE of all learners on the adapted IHDP data-set. Averaged across 100 realizations, standard error in parentheses.

Model	In-Sample	Hold-out
TNet	0.761 (0.011)	0.770 (0.013)
SNet-1	0.678 (0.009)	0.689 (0.012)
SNet-2	0.676 (0.009)	0.687 (0.012)
SNet-3	0.683 (0.009)	0.691 (0.011)
SNet	0.729 (0.009)	0.737 (0.011)
RA-Learner + TNet	0.740 (0.010)	0.745 (0.013)
RA-Learner + SNet-2	**0.670** (0.009)	**0.680** (0.012)
DR-Learner + TNet	0.893 (0.017)	0.902 (0.019)
PW-Learner + TNet	2.250 (0.093)	2.285 (0.094)

In Table 1, we observe that information sharing in plug-in learners indeed significantly improves performance versus the simple TNet, and that more complex models (SNet-3 & SNet) underperform simpler models (SNet-1 & 2). Both of these observations are not surprising given the small sample size, the high treatment group imbalance, as well as the fact that there is no true separation of covariates into different adjustment sets in the DGP. Further, we observe that – with the exception of using the RA-Learner on top of the best-performing plug-in model – the two-step learners underperform the plug-in learners on the IHDP data-set, a direct consequence of the complexity of the simulated $\tau(x)$. Further, potentially because overlap is incomplete in this data-set, the RA-learner outperforms the DR-learner substantially.

7 CONCLUSION

In this paper we considered meta-learning strategies for nonparametric CATE estimation, theoretically analyzed their properties, and implemented them using a range of neural network architectures. We demonstrated that while the DR-learner is asymptotically optimal in theory, both the RA-learner and plug-in learners sharing information between nuisance estimation tasks can outperform it in finite samples. We also showed that using sophisticated architectures as nuisance estimators for two-step learners instead of vanilla NNs can boost their small sample performance. In addition, we highlighted that the relative performance of different learners and architectures depends both on the underlying DGP and the amount of data at hand, such that the choice of learner in practice should incorporate an expert's assessment of the most likely DGP. While we considered only the choice between meta-learners using the same underlying method throughout, investigating the optimal choice of ML method – e.g. NNs versus random forests – would be an interesting next step.

[5] We noticed that the scale of CATE varies by orders of magnitude across different settings in the original data, making (R)MSE incomparable across runs. As discussed in the supplement, we rescaled the responses to correct for this.

Acknowledgements

We thank anonymous reviewers as well as members of the vanderschaar-lab for many insightful comments and suggestions. AC gratefully acknowledges funding from AstraZeneca.

References

Alaa, A. M. and van der Schaar, M. (2018a). Bayesian nonparametric causal inference: Information rates and learning algorithms. *IEEE Journal of Selected Topics in Signal Processing*, 12(5):1031–1046.

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

Alaa, A. M. and Van Der Schaar, M. (2019). Validating causal inference models via influence functions. In *International Conference on Machine Learning*, pages 191–201.

Athey, S., Tibshirani, J., and Wager, S. (2019). Generalized random forests. *Annals of Statistics*, 47(2):1179–1203.

Ballman, K. V. (2015). Biomarker: predictive or prognostic? *Journal of clinical oncology: official journal of the American Society of Clinical Oncology*, 33(33):3968–3971.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Bica, I., Alaa, A. M., Lambert, C., and van der Schaar, M. (2020). From real-world patient data to individualized treatment effects using machine learning: Current and future methods to address underlying challenges. *Clinical Pharmacology & Therapeutics.*

Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1):41–75.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., and Newey, W. (2017). Double/debiased/neyman machine learning of treatment effects. *American Economic Review*, 107(5):261–65.

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

Curth, A., Alaa, A. M., and van der Schaar, M. (2020). Estimating structural target functions using machine learning and influence functions. *arXiv preprint arXiv:2008.06461.*

Fan, Q., Hsu, Y.-C., Lieli, R. P., and Zhang, Y. (2020). Estimation of conditional average treatment effects with high-dimensional data. *Journal of Business & Economic Statistics*, pages 1–15.

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.

Horvitz, D. G. and Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685.

Imbens, G. W. (2004). Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics*, 86(1):4–29.

Johansson, F., Shalit, U., and Sontag, D. (2016). Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029.

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

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.

Lee, S., Okui, R., and Whang, Y.-J. (2017). Doubly robust uniform confidence band for the conditional average treatment effect function. *Journal of Applied Econometrics*, 32(7):1207–1225.

Nie, X. and Wager, S. (2020). Quasi-oracle estimation of heterogeneous treatment effects. *arXiv preprint arXiv:1712.04912.*

Robins, J. M. and Rotnitzky, A. (1995). Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association*, 90(429):122–129.

Robinson, P. M. (1988). Root-n-consistent semiparametric regression. *Econometrica: Journal of the Econometric Society*, pages 931–954.

Rolling, C. A. and Yang, Y. (2014). Model selection for estimating treatment effects. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pages 749–769.

Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of* *the American Statistical Association*, 100(469):322–331.

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.

van der Laan, M. J. and Rose, S. (2011). *Targeted learning: causal inference for observational and experimental data*. Springer Science & Business Media.

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*.

Yoon, J., Jordon, J., and van der Schaar, M. (2018). Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*.

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.