

Article

A Meta-Learning Approach for Estimating Heterogeneous Treatment Effects Under Hölder Continuity

Zhihao Zhao  and Congyang Zhou *

School of Statistics, Capital University of Economics and Business, Beijing 100070, China; zhzhao@cueb.edu.cn

* Correspondence: 22023120118@cueb.edu.cn

Abstract: Estimating heterogeneous treatment effects plays a vital role in many statistical applications, such as precision medicine and precision marketing. In this paper, we propose a novel meta-learner, termed RXlearner for estimating the conditional average treatment effect (CATE) within the general framework of meta-algorithms. RXlearner enhances the weighting mechanism of the traditional Xlearner to improve estimation accuracy. We establish non-asymptotic error bounds for RXlearner under a continuity classification criterion, specifically assuming that the response function satisfies Hölder continuity. Moreover, we show that these bounds are achievable by selecting an appropriate base learner. The effectiveness of the proposed method is validated through extensive simulation studies and a real-world data experiment.

Keywords: conditional average treatment effect; heterogeneous treatment effect; causal inference; minimax optimality; Hölder continuous

MSC: 62-08; 62P99



Academic Editor: Charlotte Wang

Received: 2 May 2025

Revised: 17 May 2025

Accepted: 23 May 2025

Published: 24 May 2025

Citation: Zhao, Z.; Zhou, C.

A Meta-Learning Approach for Estimating Heterogeneous Treatment Effects Under Hölder Continuity.

Mathematics **2025**, *13*, 1739. <https://doi.org/10.3390/math13111739>**Copyright:** © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license

(<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Causal inference plays a pivotal role across a wide range of scientific disciplines. With the rapid advancement of big data technologies, researchers now have access to more information than ever before, enabling more accurate estimation of causal effects. A central task in this field is the estimation of heterogeneous treatment effects (HTEs), which seeks to quantify how treatment effects vary across individuals or subpopulations. This is particularly critical in domains such as precision medicine and targeted policy-making, where practitioners—such as doctors, policymakers, and researchers—aim to determine whether newly developed treatments or interventions produce the desired outcomes for different groups.

As research in causal inference has progressed, it has become increasingly clear that average treatment effects (ATEs) may mask significant heterogeneity at the individual level. As a result, growing attention has been devoted to the estimation of conditional average treatment effects (CATEs), which aim to capture individual-level causal effects conditional on observed covariates.

Early approaches to estimating CATEs include semi-parametric models, such as partial linear models [1,2] and additive models [3], as well as classical non-parametric methods [4]. In addition, several weighting-based methods have been proposed, including inverse probability weighting (IPW), augmented IPW (AIPW) [5–7], and propensity score optimization techniques [8]. These traditional methods are supported by mature theoretical foundations but often rely on restrictive modeling assumptions, which limit their flexibility in capturing complex real-world data structures.

With the rise of machine learning, researchers have developed a variety of flexible, data-driven methods for estimating CATEs. Ref. [9] introduced a model-free meta-learning algorithm, which was further extended by [10] through a general framework that includes the Slearner and Tlearner [11,12], and later the Xlearner. Other notable contributions include the Rlearner based on Robinson decomposition [13], Athey’s Causal Forest [14], and the Double Machine Learning (DML) framework [15], which utilizes Neyman orthogonality and cross-fitting to reduce sensitivity to nuisance parameters. The Doubly Robust Learner (DRL) [16] and the unifying framework proposed by [17]—which incorporates the $L1$ loss to enhance robustness—further broaden the scope of meta-learning-based estimators.

These modern approaches leverage the flexibility of machine learning to model complex functional relationships and have demonstrated strong empirical performance. However, despite their practical success, most theoretical analyses of these methods rely on strong smoothness assumptions, such as requiring the response function to lie in a reproducing kernel Hilbert space (RKHS) or satisfy Lipschitz continuity. While these assumptions facilitate theoretical derivations, they may not hold in many real-world applications.

Among the existing meta-learning methods, the Xlearner has demonstrated strong empirical performance, particularly in scenarios with covariate unbalance and unequal treatment assignment rates. Its design leverages different base learners for treated and control groups, which allows it to adapt well to unbalanced datasets and heterogeneous response surfaces [10]. These features make Xlearner one of the most widely used and practically effective methods in CATE estimation. However, despite its strengths, Xlearner still faces two main challenges. First, as noted by [10], its reliance on propensity scores as weighting functions can limit its flexibility, particularly in complex or high-dimensional data settings. Second, most of its theoretical guarantees rely on the assumption that the underlying response functions are Lipschitz continuous, which may not hold in many practical scenarios where the functions are less smooth or exhibit discontinuities.

To address the limitations of fixed-weight designs and the error accumulation inherent in the Xlearner, while preserving its structural advantages, we propose a novel meta-learning method called RXlearner. This method incorporates a data-driven, covariate-dependent weighting mechanism that adaptively combines pseudo-treatment effect estimates. By doing so, RXlearner enhances the model’s flexibility in capturing complex response patterns and mitigates cumulative errors across estimation stages.

We rigorously establish the theoretical properties of RXlearner by deriving a non-asymptotic error bound under the assumption that the response function satisfies Hölder continuity, a milder and more general condition than those typically assumed in the existing literature. The effectiveness of RXlearner is further demonstrated through extensive simulation studies and a real-world application. While the inherent variability in the data makes the identification of a universally optimal estimator elusive, our results show that RXlearner delivers consistently competitive and robust performance across a broad range of scenarios.

The remainder of this paper is organized as follows. Section 2 introduces the model assumptions and presents the RXlearner algorithm. Section 3 develops the theoretical analysis of the estimator, where we derive non-asymptotic error bounds under Hölder continuity. To evaluate the practical effectiveness of the proposed method, Section 4 reports results from simulation studies, and Section 5 applies RXlearner to a real-world dataset. Section 6 concludes the paper with further discussions and directions for future research. Technical proofs are provided in Appendix A.

2. Methodology

2.1. Models and Assumptions

We consider the estimation of the conditional average treatment effect (CATE) under the potential outcomes framework of Neyman–Rubin [18,19]. Let $W \in \{0, 1\}$ be a binary treatment indicator, and for each unit i , let Y_i^1 and Y_i^0 denote the potential outcomes under treatment and control, respectively. Let $X_i \in \mathcal{R}^p$ represent the p -dimensional covariates. We assume that the data are generated independently from a distribution $\mathcal{P} \in \mathcal{D}$, such that

$$(Y_i^1, Y_i^0, X_i, W_i) \sim \mathcal{P}.$$

Under the Stable Unit Treatment Value Assumption (SUTVA) [20], the observed outcome is given by

$$Y_i = \mathbb{I}(W_i = 1)Y_i^1 + \mathbb{I}(W_i = 0)Y_i^0,$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. The observed dataset is denoted as

$$\mathcal{D} = \{(Y_i, X_i, W_i)\}_{i=1}^N.$$

We define the response functions as

$$\mu_1(x) = \mathbb{E}[Y^1 \mid X = x], \quad \mu_0(x) = \mathbb{E}[Y^0 \mid X = x],$$

and the corresponding conditional average treatment effect (CATE) function is

$$\tau(x) = \mathbb{E}[Y^1 - Y^0 \mid X = x] = \mu_1(x) - \mu_0(x).$$

To simulate observed outcomes, we further adopt an additive noise model

$$Y_i = \mu_{W_i}(X_i) + \sigma\epsilon_i.$$

The variance parameter σ controls the noise level and is varied across simulation scenarios.

The propensity score, i.e., the probability of receiving treatment conditional on covariates, is defined as

$$e(x) = \mathbb{P}(W = 1 \mid X = x). \quad (1)$$

Our goal is to estimate τ with an estimator $\hat{\tau}$, and to evaluate its performance under the expected mean squared error (EMSE), defined as

$$\text{EMSE}(\mathcal{P}, \hat{\tau}) = \mathbb{E}[(\tau(\mathcal{X}) - \hat{\tau}(\mathcal{X}))^2].$$

Here, we follow the EMSE definition proposed in [10], where $\mathcal{X} \sim \Lambda$ and $\mathcal{X}$ is assumed to be independent of $\hat{\tau}$. In our implementation, this assumption is addressed via a sample-splitting strategy: the estimator $\hat{\tau}$ is trained on one subset of the data, and EMSE is evaluated on an independent test sample drawn from the same distribution. This design ensures that $\hat{\tau}(\mathcal{X})$ is independent of $\mathcal{X}$ at evaluation time. Consequently, Λ corresponds to the marginal distribution of the evaluation sample.

To establish the theoretical properties of the estimator, we make the following standard assumptions.

Assumption 1 (Ignorability). *The treatment assignment W_i is independent of the potential outcomes (Y_i^1, Y_i^0) conditional on the covariates X_i , i.e.,*

$$(Y_i^1, Y_i^0) \perp W_i \mid X_i.$$

Assumption 2 (Positivity). *The propensity score is bounded away from 0 and 1, i.e., $e(x) \in (0, 1)$ for all $x \in \mathcal{X}$.*

Assumption 3 (Conditionally Independent Errors). *The errors are independent of treatment assignment given the covariates, i.e., $\epsilon_i \perp W_i \mid X_i$. We further assume that $\mathbb{E}[\epsilon_i \mid X_i] = 0$ and that the conditional variance of the errors exists.*

Remark 1. *Assumption 3 states that the error term ϵ_i is conditionally independent of the treatment assignment W_i given the covariates X_i . This assumption ensures that the estimation of nuisance functions (such as the outcome regressors and the imputed treatment effect differences in the first stage) is unbiased. It is a standard condition in many meta-learner frameworks, including the Xlearner. In our RXlearner, this assumption supports the consistency of the refined weighting strategy. While the assumption may be restrictive in practice, it provides a clean theoretical foundation. We acknowledge this limitation and leave the relaxation of this assumption for future exploration.*

2.2. Meta-Algorithms

In this section, we begin by reviewing representative models within the meta-learner framework for CATE estimation. Meta-learners are a class of model-agnostic methods that reduce the problem of causal effect estimation to a series of supervised learning tasks, enabling the use of flexible machine learning algorithms as base learners. We briefly describe three widely used approaches: the Slearner, Tlearner, and Xlearner, which form the foundation for our proposed RXlearner.

The Slearner estimates treatment effects using a single predictive model, where “S” stands for “single”. It incorporates the treatment indicator as one of the input features, treating it on an equal footing with other covariates. The response function is modeled as

$$\mu(x, w) = \mathbb{E}[Y \mid X = x, W = w],$$

where $w \in \{0, 1\}$.

The estimated conditional average treatment effect (CATE) at the covariate value x is then given by

$$\hat{\tau}_S(x) = \hat{\mu}(x, 1) - \hat{\mu}(x, 0).$$

While the Slearner uses a single model incorporating the treatment indicator, the Tlearner fits two models independently for each treatment group.

The Tlearner estimates treatment effects by splitting the dataset into treated and control groups and fitting separate models to each subgroup. The “T” stands for “two”, reflecting the use of two distinct models, one for each treatment condition.

The response function for the treated group is modeled as

$$\mu_1(x) = \mathbb{E}[Y^1 \mid X = x], \quad (2)$$

and for the control group as

$$\mu_0(x) = \mathbb{E}[Y^0 \mid X = x]. \quad (3)$$

The estimated CATE is then computed as the difference between the two fitted models:

$$\hat{\tau}_T(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x).$$

A limitation of the Tlearner is that it estimates treatment effects separately for each group without borrowing strength from the other. The Xlearner improves upon this by incorporating cross-group information through imputation.

The Xlearner builds upon the Tlearner and proceeds in three main steps. The first step mirrors the Tlearner: the response functions for the treated and control groups, $\mu_1(x)$ and $\mu_0(x)$, are estimated as in Equations (2) and (3).

In the second step, pseudo-treatment effects are imputed by leveraging the estimated response functions from the opposite group. Specifically, the imputed treatment effects are computed as

$$\tilde{D}_i^1 := Y_i^1 - \hat{\mu}_0(X_i^1), \quad \tilde{D}_i^0 := \hat{\mu}_1(X_i^0) - Y_i^0. \quad (4)$$

These imputed differences are then used to estimate the treatment effects conditional on covariates:

$$\tau_1(x) := \mathbb{E}[\tilde{D}^1 \mid X = x], \quad \tau_0(x) := \mathbb{E}[\tilde{D}^0 \mid X = x]. \quad (5)$$

If the response functions are correctly estimated, i.e., $\hat{\mu}_0 = \mu_0$ and $\hat{\mu}_1 = \mu_1$, then

$$\tau(x) = \mathbb{E}[\tilde{D}^1 \mid X = x] = \mathbb{E}[\tilde{D}^0 \mid X = x].$$

Any supervised learning or regression method can be employed to estimate $\tau(x)$ by regressing the imputed treatment effects on the covariates within each treatment arm. This yields two estimators: $\hat{\tau}_1(x)$ from the treated group and $\hat{\tau}_0(x)$ from the control group.

In the third step, the final CATE estimate is obtained by combining the two pseudo-effect estimates using a weighting function $g(x) \in [0, 1]$:

$$\hat{\tau}_X(x) = g(x)\hat{\tau}_0(x) + (1 - g(x))\hat{\tau}_1(x),$$

where $g(x)$ typically depends on the propensity score, e.g., $g(x) = e(x)$ as in [10].

2.3. RXlearner

To improve upon the standard Xlearner framework while retaining its structural advantages, we propose a novel meta-learning method termed **RXlearner** (Refitting Xlearner). This method enhances the traditional weighting strategy by incorporating a data-driven mechanism to adaptively combine the two pseudo-treatment effect estimators, $\tau_0(x)$ and $\tau_1(x)$, based on covariate information.

The procedure begins with a standard Xlearner step, wherein we estimate $\tau_0(x)$ and $\tau_1(x)$ on the training set, as well as the propensity score $e(x)$. Using these quantities, we construct a pseudo-response variable:

$$\tau_r(x) := e(x) \cdot \tau_0(x) + (1 - e(x)) \cdot \tau_1(x),$$

which serves as a proxy for the unobserved individual-level treatment effect.

Instead of relying on the fixed weight function $g(x) = e(x)$ as in the original Xlearner, we adopt a two-stage refitting strategy. In the first stage, we fit a powerful regression model (e.g., Random Forest) to predict $\tau_r(x)$ using x as input, effectively capturing complex nonlinear relationships between covariates and treatment heterogeneity. However, direct

use of this refitted model may lead to instability, especially when the learned $\tau_r(x)$ is noisy in regions with limited overlap.

To mitigate this, we employ a second stage that recasts the refitted predictions into a convex combination form, aligning with the Xlearner's structure. Specifically, we aim to learn a new weighting function $g(x)$ by minimizing the squared error between the first-stage prediction $\hat{\tau}_{\text{refit}}(x)$ and a convex combination of $\tau_0(x)$ and $\tau_1(x)$:

$$\hat{g}(x) := \arg \min_g \sum_{x \in \mathcal{D}_{\text{test}}} (\hat{\tau}_{\text{refit}}(x) - [g(x) \cdot \tau_0(x) + (1 - g(x)) \cdot \tau_1(x)])^2, \quad (6)$$

where $\hat{\tau}_{\text{refit}}(x)$ denotes the pseudo-treatment effect from the refitting step, and $\mathcal{D}_{\text{test}}$ denotes the test set.

To approximate $\hat{g}(x)$, we frame this optimization as a supervised regression problem and adopt gradient boosted regression trees (GBRT) [21] implemented via the xgboost package. GBRT minimizes squared loss through functional gradient descent, providing explicit convergence guarantees under standard conditions and offering strong empirical stability in practice.

The final estimate of the CATE is then given by the following:

$$\hat{\tau}(x) = \hat{g}(x) \cdot \tau_0(x) + (1 - \hat{g}(x)) \cdot \tau_1(x). \quad (7)$$

Remark 2. It is important to note that the learned weighting function $\hat{g}(x)$ is defined on the test set $\mathcal{D}_{\text{test}}$, rather than on the entire covariate space $\mathcal{X}$. This is because the optimization objective in Equation (6) is constructed only over the test points.

Remark 3. To construct pseudo-outcomes, we apply propensity-score-based weighting similar to the Xlearner framework [10], assigning weights $1 - \hat{e}(X)$ to treated units and $\hat{e}(X)$ to control units. However, our RXlearner differs from the standard Xlearner in that the weight function $g(x)$ is not fixed to $\hat{e}(x)$. Instead, it is refined through a second-stage regression step, where the optimal weights are learned in a data-driven manner by minimizing a squared loss objective (Equation (6)). This allows RXlearner to adaptively learn context-specific weights that may improve performance.

This two-stage design leverages the flexibility of machine learning models to fit pseudo-response values while preserving the interpretable structure of the Xlearner. Specifically, RXlearner can be viewed as a data-driven generalization of the Xlearner: instead of using the propensity score $e(x)$ as a fixed weighting function, it learns a flexible weight $g(x)$ by minimizing a squared loss with respect to a pseudo-target. This allows RXlearner to adaptively combine the two pseudo-treatment effect estimators $\tau_0(x)$ and $\tau_1(x)$, depending on their relative reliability across the covariate space. Such adaptivity improves estimation robustness and accuracy, especially in the presence of covariate imbalance or heterogeneous noise, making RXlearner well-suited for complex causal inference tasks.

We emphasize that RXlearner can be viewed as a data-driven generalization of Xlearner. In particular, when the optimization problem in (6) is solved with a fixed weight function $g(x) = e(x)$, the RXlearner reduces exactly to the original Xlearner formulation. That is, if the refitted model $\hat{\tau}_{\text{refit}}(x)$ perfectly recovers the pseudo-response $\tau_r(x) = e(x) \cdot \tau_0(x) + (1 - e(x)) \cdot \tau_1(x)$, then the optimal solution to (6) is attained when $g(x) = e(x)$. This establishes the Xlearner as a special case of RXlearner.

To facilitate implementation, we summarize the RXlearner procedure in Algorithm 1 as follows.

Algorithm 1 RXlearner Algorithm

- 1: **Input:** Observed dataset $\mathcal{D} = \{(X_i, Y_i, W_i)\}_{i=1}^n$
- 2: **Output:** Estimated conditional average treatment effect $\hat{\tau}(x)$ on the test set
- 3: **Step 0: Data Splitting**
- 4: Randomly split $\mathcal{D}$ into training set $\mathcal{D}_{\text{train}}$ and test set $\mathcal{D}_{\text{test}}$
- 5: **Step 1: Estimate Nuisance Functions on Training Set**
- 6: Use $\mathcal{D}_{\text{train}}$ to estimate the following:
 - Propensity score $\hat{e}(x) = \mathbb{P}(W = 1 \mid X = x)$
 - Conditional response functions $\hat{\mu}_0(x) = \mathbb{E}[Y \mid X = x, W = 0]$, $\hat{\mu}_1(x) = \mathbb{E}[Y \mid X = x, W = 1]$
- 7: **Step 2: Construct Pseudo-Treatment Effects on Training Set**
- 8: For units in the treated group: $\tau_1(x) := Y - \hat{\mu}_0(x)$
- 9: For units in the control group: $\tau_0(x) := \hat{\mu}_1(x) - Y$
- 10: **Step 3: Construct Pseudo-Target $\tau_r(x)$**
- 11: Combine the above using the estimated propensity score:

$$\tau_r(x) := \hat{e}(x) \cdot \tau_0(x) + (1 - \hat{e}(x)) \cdot \tau_1(x)$$

- 12: **Step 4: Learn Weighting Function $\hat{g}(x)$ by (6).**
- 13: **Step 5: Compute Final CATE on Test Set**
- 14: On $\mathcal{D}_{\text{test}}$, compute:

$$\hat{\tau}(x) = \hat{g}(x) \cdot \hat{\tau}_0(x) + (1 - \hat{g}(x)) \cdot \hat{\tau}_1(x)$$

- 15: **return** $\hat{\tau}(x)$ for $x \in \mathcal{D}_{\text{test}}$

Remark 4. To ensure reproducibility, we specify the base regression models and tuning parameters used in our implementation of the RXlearner. For the pseudo-target refitting model $\hat{\tau}_r(x)$ in Step 3, we use a random forest regressor (R package: `randomForest`, `ntree = 100`). For learning the weighting function $\hat{g}(x)$ in Step 4, we adopt gradient boosted regression trees (GBRT) using the `xgboost` package, with squared error loss, `nrounds = 100`, `max_depth = 3`, and `learning_rate = 0.1`. These choices balance flexibility and convergence stability across all experimental settings.

3. Asymptotic Properties

Assuming that only the weighting function is altered, the convergence analysis of RXlearner largely parallels that of Xlearner. Reference [10] established the convergence rate of the Xlearner under the assumption that the response function satisfies Lipschitz continuity. In this work, we generalize this assumption to Hölder continuity and derive the corresponding convergence rate for the RXlearner under this broader condition.

3.1. Fundamental Definitions and Results

To proceed, we begin by reviewing several fundamental definitions and results in the minimax nonparametric regression literature. Definition 1 introduces the concept of Hölder continuity, while Definitions 2 and 3, and Lemma 1 are adapted from [22].

It is worth noting that Hölder continuity provides a broader notion of smoothness than Lipschitz continuity, allowing for a controlled, potentially nonlinear rate of change in the function.

Definition 1. Let $f : X \rightarrow \mathcal{R}$ be a function defined on a metric space X . f is referred to as Hölder continuous if there exist constants $L > 0$ and $\alpha \in (0, 1]$ such that for any $x_1, x_2 \in X$, the following inequality holds:

$$|f(x_1) - f(x_2)| \leq L \|x_1 - x_2\|^\alpha.$$

The constant L is called the Hölder constant, and α is called the Hölder exponent. When $\alpha = 1$, Hölder continuity reduces to Lipschitz continuity.

Definition 2. Let $p = k + \beta$ under the conditions $k \in \mathcal{N}_0$ and $0 < \beta \leq 1$, and let $C > 0$. A function $f : \mathcal{R}^d \rightarrow \mathcal{R}$ is called (p, C) -smooth if, for every $\alpha = (\alpha_1, \dots, \alpha_d)$, where $\alpha_i \in \mathcal{N}_0$ and $\sum_{j=1}^d \alpha_j = k$, the k -th order partial derivative $\frac{\partial^k f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}$ exists and satisfies

$$\left| \frac{\partial^k f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}(x) - \frac{\partial^k f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}(z) \right| \leq C \cdot \|x - z\|^\beta \quad (x, z \in \mathcal{R}^d).$$

Let $\mathcal{F}^{(p,C)}$ denote the collection of all (p, C) -smooth functions $f : \mathcal{R}^d \rightarrow \mathcal{R}$. When $k = 0$, the functions in $\mathcal{F}^{(p,C)}$ as defined in Definition 2 satisfy the Hölder continuity condition.

Definition 3. Let $\mathcal{D}^{(p,C)}$ be a class of distributions on (X, Y) such that:

1. The features X_i are identically distributed in $[0, 1]^d$;
2. $Y = m(X) + N$, where X and N are independent, and N follows a standard normal distribution;
3. $m \in \mathcal{F}^{(p,C)}$.

In the Lemma 1, we derive a lower minimax rate of convergence for this class of distributions.

Lemma 1. For the class $\mathcal{D}^{(p,C)}$, the sequence

$$a_n = n^{-\frac{2p}{2p+d}}$$

is the minimax lower bound rate of convergence. Specifically, for some constant C_1 independent of C ,

$$\liminf_{n \rightarrow \infty} \inf_{m_n} \sup_{(X,Y) \in \mathcal{D}^{(p,C)}} \frac{\mathbb{E}\{\|m_n - m\|^2\}}{C^{\frac{2d}{2p+d}} n^{-\frac{2p}{2p+d}}} \geq C_1 > 0.$$

3.2. RXlearner Convergence Rate in General

To demonstrate the convergence level of the RXlearner, some preparatory work is also requisite. We give the following definitions.

Definition 4. Let F^H be a class of distributions on $(X, Y) \in [0, 1]^d \times \mathcal{R}$ such that

1. The features X_i are independent and identically distributed (i.i.d.) and uniformly distributed in $[0, 1]^d$;
2. The observed outcomes are $Y_i = \mu(X_i) + \varepsilon_i$, where each ε_i is independent and follows a normal distribution with mean 0 and variance σ^2 ;
3. X_i and ε_i are independent;
4. The response function μ is Hölder continuous with parameters L and α .

Definition 5. Let $\mathcal{D}_{mn}^H$ be a family of distributions on $(Y^0, Y^1, W, X) \in \mathcal{R}^N \times \mathcal{R}^N \times \{0, 1\}^N \times [0, 1]^{d \times N}$ such that

1. $N = m + n$;
2. The features X_i are i.i.d. and uniformly distributed in $[0, 1]^d$;
3. There are n treated units such that $\sum_i W_i = n$;
4. The observed outcomes are $Y_i(\omega) = \mu_\omega(X_i) + \varepsilon_{\omega i}$, where each $(\varepsilon_{0i}, \varepsilon_{1i})$ is independent and follows a normal distribution with mean 0 and marginal variance σ^2 ;
5. X, W , and $\varepsilon = (\varepsilon_{0i}, \varepsilon_{1i})$ are mutually independent;

6. The response functions $\mu_0(x)$ and $\mu_1(x)$ are Hölder continuous with parameters L and α .

Remark 5. We consider that for a fixed n with $0 < n < N$, we have the distribution of $(X_i, Y_i, W_i)_{i=1}^N$ given that we observe n treated units and $m = N - n$ control units. We denote this distribution by $\mathcal{P}^{mn}$.

$$\left[(X_i, Y_i, W_i)_{i=1}^N \mid \sum_{i=1}^N W_i = n \right] \sim \mathcal{P}^{mn}.$$

We note that under $\mathcal{P}^{mn}$ the (X_i, Y_i, W_i) are identical in distribution, but not independent.

Next, we will derive a lower bound on the best achievable convergence rate for $\mathcal{D}_{mn}^H$. The following theorem establishes the minimax lower bound on the rate of convergence for any estimator $\hat{\tau}^{mn}$ under Hölder continuity.

Theorem 1 (Minimax Lower Bound). Let $\hat{\tau}^{mn} \in \mathcal{D}_{mn}^H$ be an arbitrary estimator of τ , and let $a_1, a_2 > 0$ and $c > 0$ be constants such that for all $m, n \geq 1$,

$$\sup_{\mathcal{P} \in \mathcal{D}_{mn}^H} \text{EMSE}(\mathcal{P}, \hat{\tau}^{mn}) \leq c(m^{-a_1} + n^{-a_2}), \quad (8)$$

then the convergence exponents must satisfy

$$a_1, a_2 \leq \frac{2\alpha}{2\alpha + d}.$$

We prove through Theorem 1 that the optimal rate of RXlearner is $O(n^{-2\alpha/(2\alpha+d)} + m^{-2\alpha/(2\alpha+d)})$.

Proof of Theorem 1. See Appendix A.1. $\square$

3.3. The Convergence Rate of RXlearner When the Base Learner Is KNN

We now show that RXlearner attains the minimax lower bound by choosing KNN as the base learner.

Theorem 2. Let $d > 2$, and assume $(X, W, Y(0), Y(1)) \sim \mathcal{P} \in \mathcal{D}_{mn}^H$, where the response functions $\mu_0(x)$ and $\mu_1(x)$ are Hölder continuous with parameters L and α , satisfying

$$|\mu_w(x) - \mu_w(z)| \leq L\|x - z\|^\alpha,$$

where $\omega \in \{0, 1\}$, $X \sim \text{Unif}[0, 1]^d$, and $0.5 < \alpha < 1$.

Furthermore, let $\hat{\tau}^{mn}$ be the RXlearner constructed using the KNN base learner with the following specifications

1. $g \equiv 0$;
2. The first-stage base learner $\hat{\mu}_0$ for the control group is a KNN estimator with $k_0 = \left\lceil (\sigma^2/L^2)^{\frac{d}{2\alpha+d}} m^{\frac{2\alpha}{2\alpha+d}} \right\rceil$;
3. The second-stage base learner $\hat{\tau}_1$ for the treated group is a KNN estimator with $k_1 = \left\lceil (\sigma^2/L^2)^{\frac{d}{2\alpha+d}} n^{\frac{2\alpha}{2\alpha+d}} \right\rceil$.

Then, $\hat{\tau}^{mn}$ attains the minimax optimal convergence rate stated in Theorem 1. Moreover, there exists a constant $C > 0$ such that

$$\mathbb{E}\|\tau - \hat{\tau}^{mn}\|^2 \leq C\sigma^{4\alpha/(2\alpha+d)} L^{\frac{2d}{2\alpha+d}} \left(m^{-2\alpha/(2\alpha+d)} + n^{-2\alpha/(2\alpha+d)} \right).$$

Proof of Theorem 2. See Appendix A.2. $\square$

4. Simulation

In this section, we consider several settings for the data-generating process, where the generation of X_i follows the approach of [17]. Here, p denotes the dimension of covariates:

$$X_i \sim \mathcal{N}_p(\mathbf{0}, \Sigma), \quad \text{diag}(\Sigma) = 1, \quad \text{Corr}(X_{ij}, X_{ik}) = \rho^{|j-k|}, \quad i = 1, \dots, n,$$

$$e_i = e(x_i), \quad w_i \sim \text{Bernoulli}(e_i),$$

$$b(X_i) = 0.5(\mu_0(X_i) + \mu_1(X_i)), \quad \tau(X_i) = \mu_1(X_i) - \mu_0(X_i),$$

$$\epsilon_i \sim \mathcal{N}(0, 1), \quad y_i = b(X_i) + (w_i - 0.5)\tau(X_i) + \sigma\epsilon_i.$$

In the following simulations, we set the training sample sizes to $n \in \{200, 500, 1000, 2000, 5000\}$, with a fixed test set size of $n_t = 10^5$, and conduct $T = 100$ independent replications for each setting. An exception is Simulation 5, which involves a highly imbalanced treatment assignment. Since a training size of $n = 200$ results in an insufficient number of treated units, we consider only training sizes from 500 to 5000 in this case.

We use MSE to evaluate the performance of each estimator. With the number of simulation replication T , we have

$$\text{MSE} = \frac{1}{T} \sum_{t=1}^T \left[\hat{\tau}^{(t)}(x_t) - \tau(x_t) \right]^2,$$

where x_t is the observed value of the test set in the t replicate, and $\hat{\tau}^{(t)}(x)$ is the estimator of $\tau(x)$ in the t th replicate.

We present our simulation results in both tabular and graphical formats. In the tables, the best-performing method under each setting is highlighted in bold for ease of comparison.

4.1. Different Response Functions

To evaluate the performance of the proposed estimator and validate the effectiveness of the theoretical framework, we first examine the performance of various estimators under different rates of change in the response function.

Simulation 1: Hölder Continuous Response Function

$$\mu_0(X_i) = 0.5|X_{i4}|^{\frac{2}{3}},$$

$$\mu_1(X_i) = 0.5|X_{i4}|^{\frac{2}{3}} + \log(1 + |X_{i5}|).$$

Simulation 2: Lipschitz Continuous Response Function

$$\mu_0(X_i) = |X_{i5}|,$$

$$\mu_1(X_i) = 0.5X_{i2} + |X_{i5}|.$$

Simulation 1 reflects the Hölder continuity structure (with exponent $\alpha = 2/3$), which aligns with the theoretical analysis in Section 3. We choose the form $\mu_0(x) = 0.5|x_{i4}|^{2/3}$ to introduce a moderate level of nonlinearity and limited smoothness, which is representative of real-world settings where the underlying structural functions are often non-differentiable or exhibit sharp curvature changes. Such Hölder-type structures have been observed in various domains, including economics and biology (e.g., Kleiber's law and related metabolic scaling theory; see [23]). This choice also allows us to examine the sensitivity

of different estimators to non-smoothness. To further assess robustness under alternative smoothness conditions, we include a Lipschitz continuous setting in Simulation 2.

For these two types of simulations, we consider $\rho = 0.5$, $e(x) = 0.5$, $p \in \{5, 10\}$ and $\sigma = \{0.5, 1, 2\}$.

4.2. Cases in Special Situations

We next examine three specially designed simulation settings to evaluate the robustness and adaptability of different estimators under practically challenging conditions. These scenarios are intended to assess the performance of our proposed RXlearner when key identification assumptions are weakened. Specifically, Simulation 3 introduces confounding by violating the unconfoundedness assumption to evaluate the estimator's robustness under model misspecification. Simulation 4 focuses on the null treatment effect case, where the true conditional treatment effect is uniformly zero across all covariate profiles, highlighting the ability of each method to avoid spurious heterogeneity. Simulation 5 investigates an extremely unbalanced treatment assignment scenario, in which the treated group constitutes a very small proportion of the sample. Given that the Xlearner is known to perform well in such settings, we aim to examine whether RXlearner can inherit or improve upon this advantage. The response function $\mu_0(X_i)$ in Simulation 3 is constructed following the specification of [24].

Simulation 3: Confounding

$$e_i = \frac{1}{1 + \exp(X_{i2} + X_{i3})},$$

$$\mu_0(X_i) = \sin(\pi X_{i1} X_{i2}) + 2(X_{i3} - 0.5)^2 + X_{i4} + 0.5X_{i5},$$

$$\mu_1(X_i) = \mu_0(X_i) + 3 \cdot I(X_{i2} > 0.1).$$

We consider $\rho = 0.5$, $p = 10$ and $\sigma = 1$.

Simulation 4: No Treatment

$$\mu_0(X_i) = \sin(\pi X_{i1} X_{i2}) + 2(X_{i3} - 0.5)^2 + X_{i4} + 0.5X_{i5},$$

$$\mu_1(X_i) = \mu_0(X_i).$$

We consider $\rho = 0.5$, $e(x) = 0.5$, $p = 10$ and $\sigma = 1$.

Simulation 5: Extremely Unbalanced Data

$$e_i = 0.01,$$

$$\mu_0(X_i) = \sin(\pi X_{i1} X_{i2}) + 2(X_{i3} - 0.5)^2 + X_{i4} + 0.5X_{i5},$$

$$\mu_1(X_i) = \mu_0(X_i) + 3 \cdot I(X_{i2} > 0.1).$$

To evaluate the tendency of each method to spuriously detect heterogeneity when the true treatment effect is absent, we compute the false positive rate (FPR) as the proportion of units with estimated CATE values exceeding a fixed threshold $\text{FPR}(\delta) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(|\hat{\tau}(X_i)| > \delta)$, where $\delta \in \{0.05, 0.2\}$ is a pre-specified constant. These thresholds reflect varying levels of tolerance for deviations from zero. The FPR results are reported in Table 1, while the MSE results of Simulations 1–5 are presented in Tables 2–4.

Table 1. False Positive Rates under Simulation 4 where the true CATE is zero.

δ	n	False Positive Rate			
		Xlearner	Slearner	Tlearner	RXlearner
0.05	200	0.927	0.553	0.957	0.906
	500	0.902	0.516	0.946	0.867
	1000	0.882	0.501	0.937	0.834
	2000	0.857	0.471	0.925	0.793
	5000	0.822	0.431	0.911	0.732
0.2	200	0.716	0.062	0.830	0.640
	500	0.626	0.049	0.786	0.509
	1000	0.557	0.034	0.752	0.416
	2000	0.478	0.024	0.709	0.315
	5000	0.378	0.016	0.658	0.201

Table 2. Simulation results of Simulation 1.

Simulation	p	σ	n	MSE			
				Xlearner	Slearner	Tlearner	RXlearner
Simulation 1	5	0.5	200	0.160	0.196	0.236	0.141
			500	0.117	0.160	0.184	0.103
			1000	0.090	0.140	0.150	0.078
			2000	0.071	0.126	0.126	0.061
			5000	0.054	0.111	0.104	0.045
		1	200	0.156	0.186	0.230	0.137
			500	0.113	0.155	0.179	0.100
			1000	0.091	0.142	0.152	0.079
			2000	0.072	0.127	0.128	0.062
			5000	0.054	0.110	0.103	0.044
		2	200	0.465	0.280	0.775	0.372
			500	0.345	0.305	0.629	0.264
			1000	0.278	0.338	0.536	0.209
			2000	0.233	0.377	0.464	0.171
			5000	0.175	0.379	0.380	0.123
	10	0.5	200	0.081	0.124	0.090	0.083
			500	0.052	0.074	0.065	0.054
			1000	0.037	0.055	0.049	0.041
			2000	0.027	0.041	0.038	0.026
			5000	0.017	0.029	0.028	0.017
		1	200	0.156	0.186	0.230	0.137
			500	0.113	0.155	0.179	0.100
			1000	0.091	0.142	0.152	0.079
			2000	0.072	0.127	0.128	0.062
			5000	0.054	0.110	0.103	0.044
		2	200	0.515	0.283	0.832	0.419
			500	0.350	0.261	0.618	0.280
			1000	0.262	0.246	0.484	0.204
			2000	0.200	0.245	0.396	0.152
			5000	0.139	0.233	0.306	0.100

Table 3. Simulation results of Simulation 2.

Simulation	p	σ	n	MSE			
				Xlearner	Slearner	Tlearner	RXlearner
Simulation 2	0.5		200	0.119	0.172	0.173	0.115
			500	0.085	0.129	0.132	0.082
			1000	0.065	0.102	0.103	0.063
			2000	0.050	0.076	0.085	0.048
			5000	0.037	0.058	0.067	0.034
		1	200	0.166	0.186	0.264	0.147
			500	0.116	0.154	0.200	0.101
			1000	0.091	0.125	0.166	0.079
			2000	0.074	0.102	0.142	0.062
			5000	0.054	0.085	0.113	0.044
	1	2	200	0.502	0.249	0.860	0.402
			500	0.350	0.214	0.646	0.268
			1000	0.283	0.197	0.558	0.209
			2000	0.229	0.183	0.475	0.164
			5000	0.171	0.187	0.387	0.116
		0.5	200	0.100	0.205	0.136	0.099
			500	0.061	0.166	0.093	0.061
			1000	0.041	0.116	0.068	0.041
			2000	0.029	0.082	0.053	0.029
			5000	0.018	0.052	0.038	0.018
	10	1	200	0.166	0.186	0.264	0.147
			500	0.116	0.154	0.200	0.101
			1000	0.091	0.125	0.166	0.079
			2000	0.074	0.102	0.142	0.062
			5000	0.054	0.085	0.113	0.044
		2	200	0.510	0.244	0.853	0.416
			500	0.348	0.219	0.648	0.272
			1000	0.250	0.202	0.500	0.187
			2000	0.194	0.185	0.421	0.141
			5000	0.137	0.151	0.330	0.094

Table 4. Simulation results of Simulations 3–5.

Simulation	n	MSE			
		Xlearner	Slearner	Tlearner	RXlearner
Simulation 3	200	1.596	2.467	4.715	1.416
	500	0.741	1.868	3.662	0.610
	1000	0.406	1.405	2.931	0.302
	2000	0.261	1.169	2.492	0.178
	5000	0.157	0.926	1.946	0.096
Simulation 4	200	0.435	0.011	1.307	0.239
	500	0.266	0.010	0.882	0.137
	1000	0.189	0.008	0.656	0.094
	2000	0.127	0.007	0.446	0.061
	5000	0.077	0.005	0.299	0.037
Simulation 5	500	3.220	3.509	9.430	3.199
	1000	2.101	3.331	8.115	2.079
	2000	1.332	3.227	7.546	1.307
	5000	0.647	2.766	5.296	0.623

4.3. Cases with Different Correlations and Noise

We further consider two scenarios: one concerning the impact of correlations among variables, and the other involving variations in the noise term.

Simulation 6: Different variable correlations and noise terms

$$\mu_0(X_i) = \sin(\pi X_{i1} X_{i2}) + 2(X_{i3} - 0.5)^2 + X_{i4} + 0.5X_{i5},$$

$$\mu_1(X_i) = \mu_0(X_i) + 3 \cdot I(X_{i2} > 0.1).$$

We consider $\rho \in \{0, 0.5\}$, $e(x) = 0.5$, $p = 10$ and $\sigma \in \{0.5, 1, 2\}$. The corresponding MSE results are summarized in Table 5.

Table 5. Simulation results of Simulation 6.

Simulation	p	σ	n	MSE			
				Xlearner	Slearner	Tlearner	RXlearner
Simulation 6	0	0.5	200	0.937	2.190	1.802	0.848
			500	0.427	1.247	1.050	0.359
			1000	0.231	0.754	0.729	0.172
			2000	0.131	0.456	0.489	0.087
			5000	0.079	0.286	0.343	0.047
		1	200	1.052	2.367	1.901	0.935
			500	0.489	1.342	1.196	0.392
			1000	0.279	0.802	0.827	0.201
			2000	0.175	0.525	0.594	0.114
			5000	0.106	0.318	0.411	0.063
		2	200	1.451	2.746	2.600	1.240
			500	0.779	1.683	1.679	0.622
			1000	0.489	1.082	1.245	0.347
			2000	0.315	0.717	0.929	0.204
			5000	0.206	0.463	0.676	0.121
	0.5	0.5	200	0.941	2.068	1.848	0.811
			500	0.400	1.117	1.093	0.323
			1000	0.225	0.655	0.720	0.168
			2000	0.131	0.433	0.488	0.090
			5000	0.073	0.265	0.325	0.045
		1	200	0.995	2.081	1.882	0.839
			500	0.466	1.140	1.149	0.372
			1000	0.281	0.766	0.822	0.207
			2000	0.178	0.499	0.603	0.118
			5000	0.103	0.316	0.397	0.061
		2	200	1.486	2.450	2.672	1.239
			500	0.752	1.559	1.682	0.572
			1000	0.502	1.063	1.241	0.362
			2000	0.325	0.692	0.946	0.212
			5000	0.211	0.496	0.669	0.125

4.4. Summary of the Results

To visually compare the performances of different methods across all simulation settings, the corresponding results are presented below in Figures 1–7.

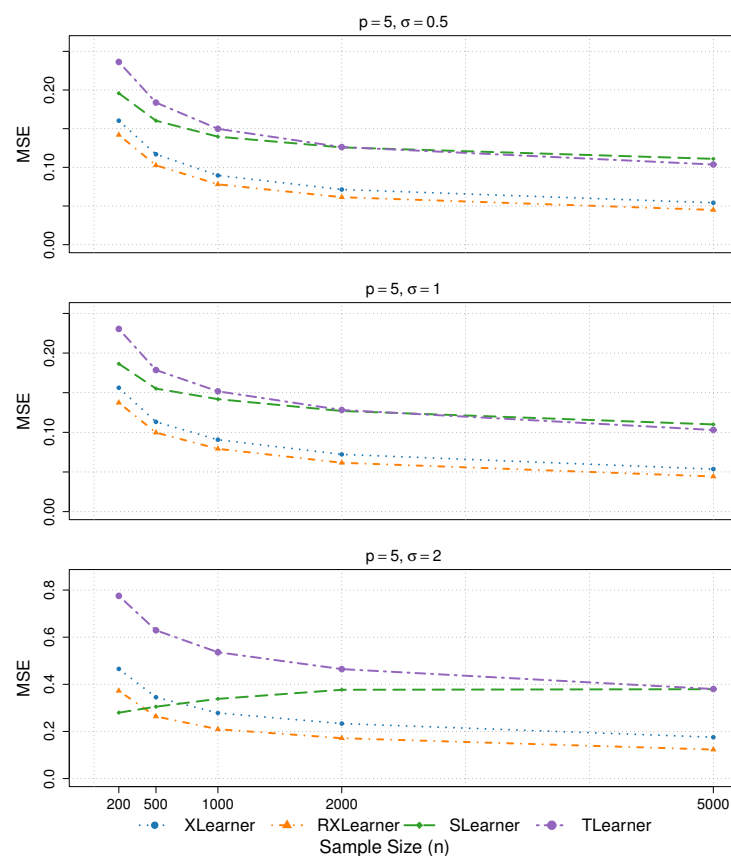


Figure 1. The results of Simulation 1 are presented, with $p = 5$.

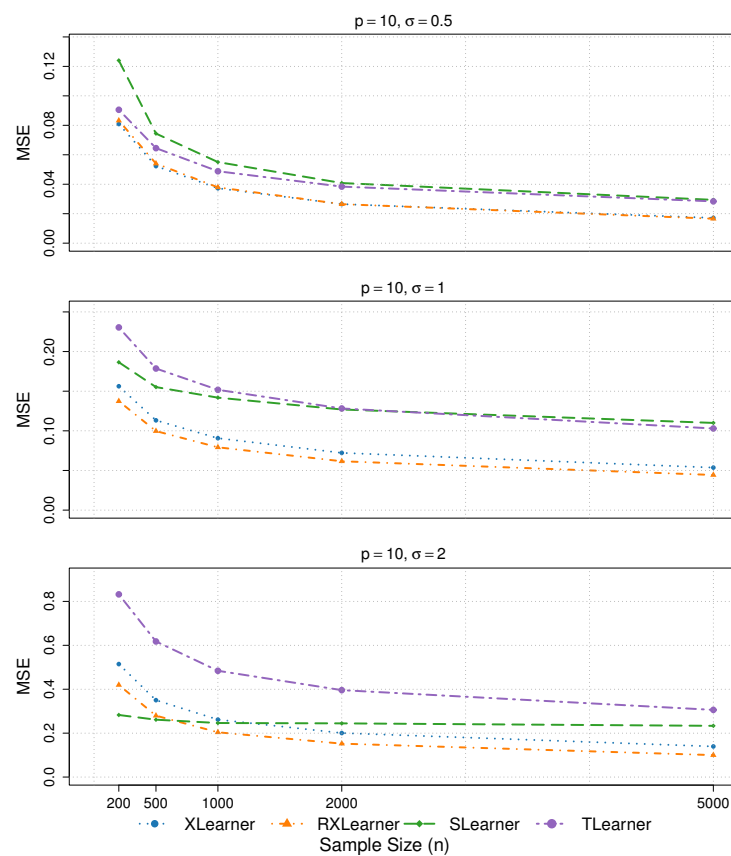


Figure 2. The results of Simulation 1 are presented, with $p = 10$.

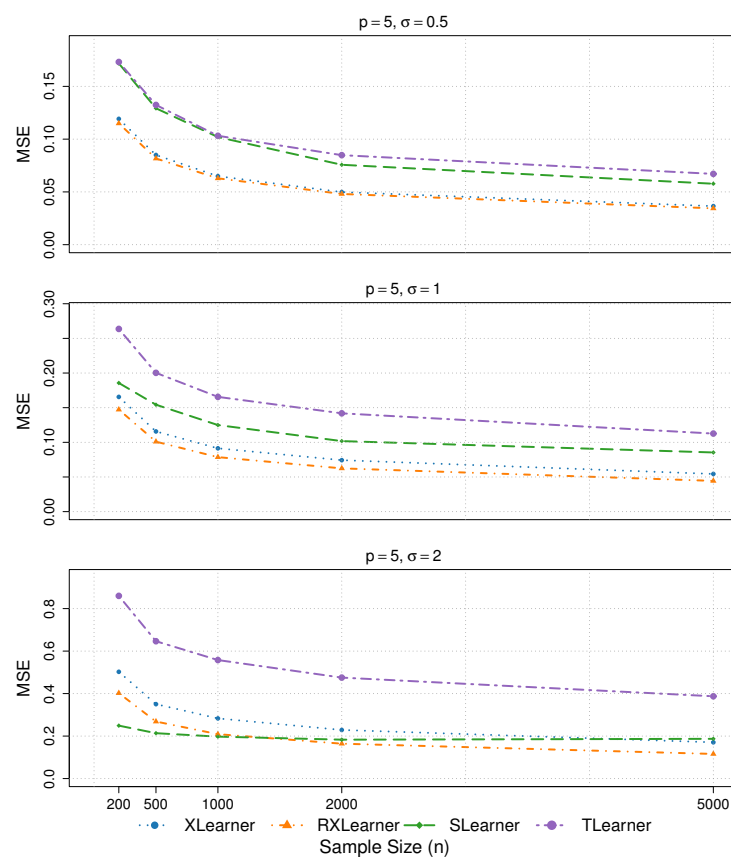


Figure 3. The results of Simulation 2 are presented, with $p = 5$.

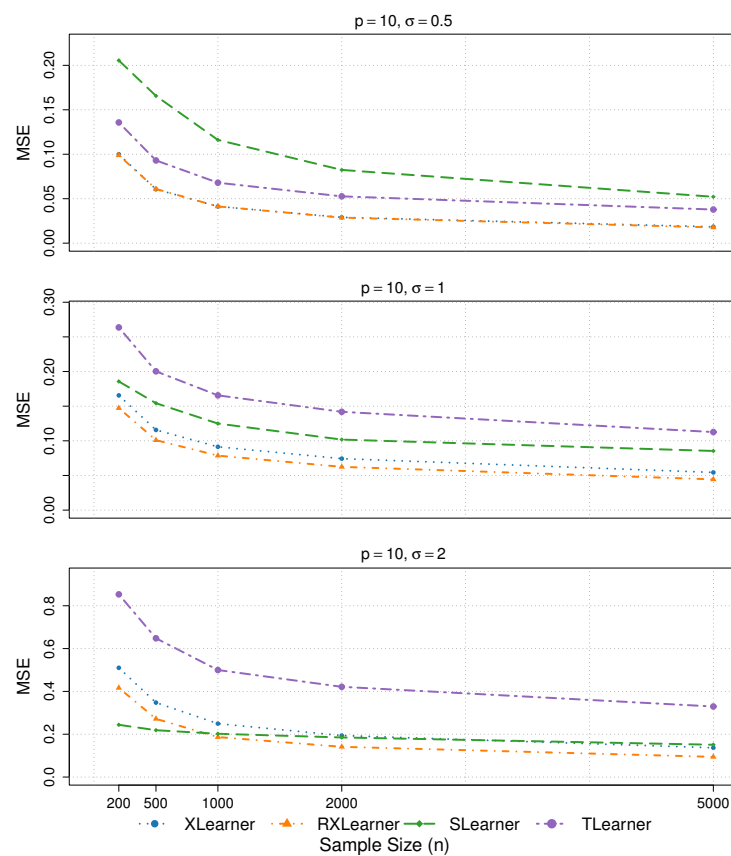


Figure 4. The results of Simulation 2 are presented, with $p = 10$.

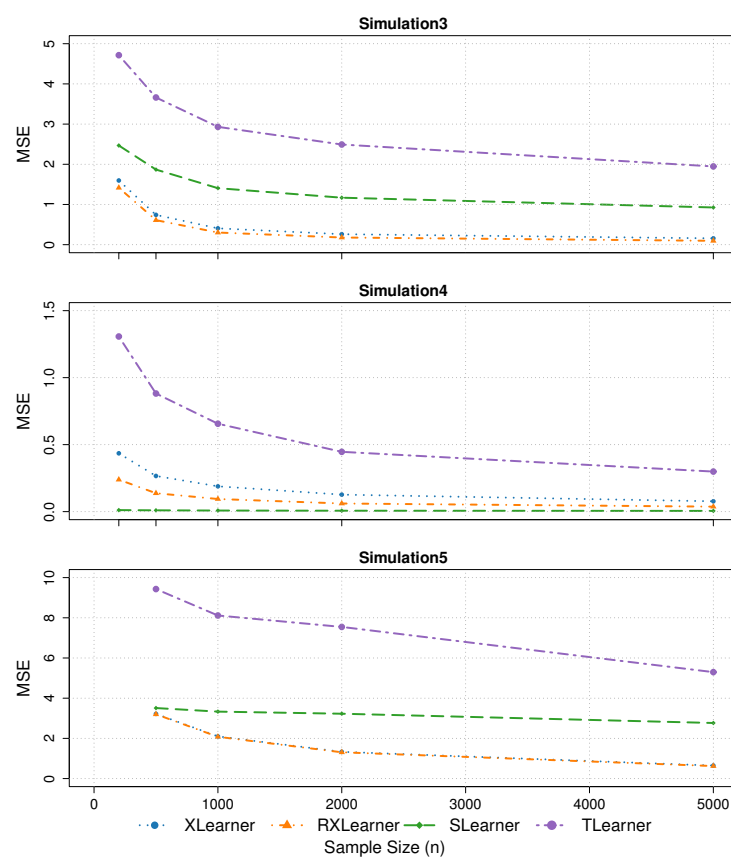


Figure 5. Results of Simulations 3–5.

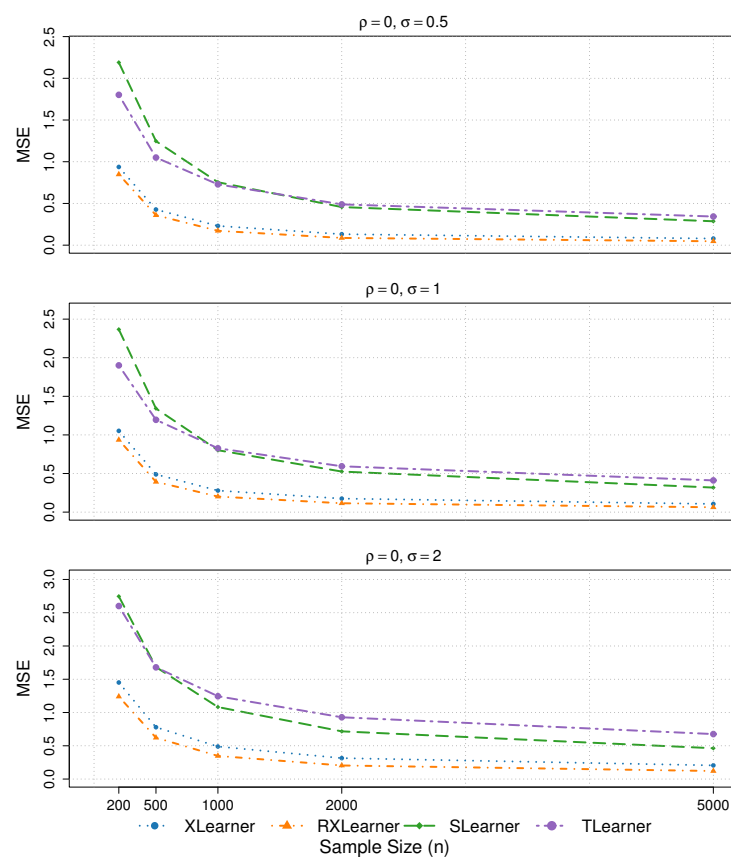


Figure 6. The results of Simulation 6 are presented, with $\rho = 0$.

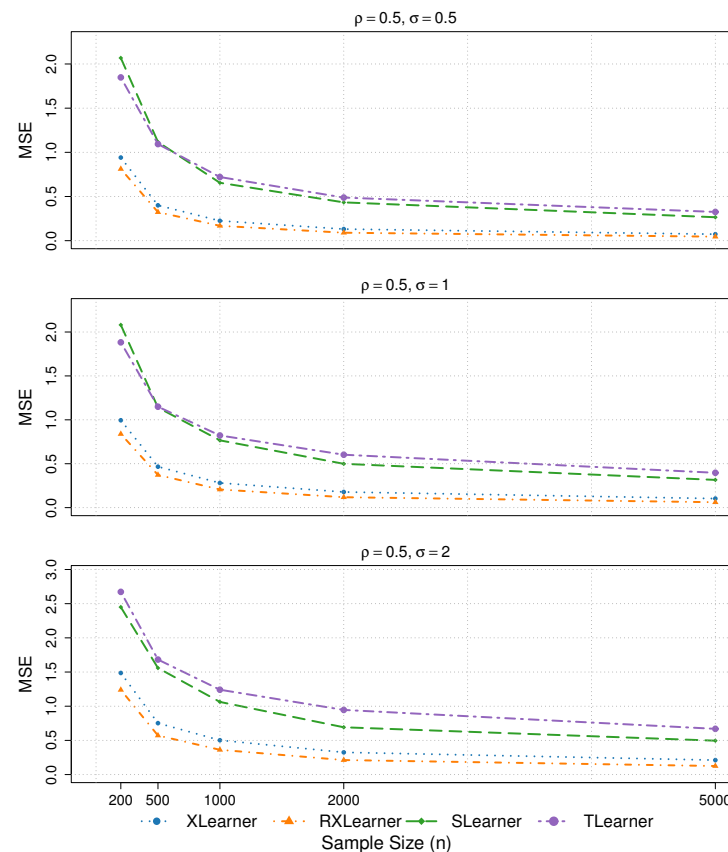


Figure 7. The results of Simulation 6 are presented, with $\rho = 0.5$.

Based on the simulations discussed above and the empirical results shown in the figures and tables, we summarize the key findings as follows:

1. In Simulations 1 and 2, both the XLearner and RXLearner outperform the SLearner and TLearner, with RXLearner achieving the best overall performance. Simulation results consistently demonstrate the superior performance of RXLearner across various noise levels ($\sigma = 0.5, 1, 2$), covariate dimensions ($p = 5, 10$), and sample sizes ($n = 200$ to 5000). As the sample size increases, all methods improve, but RXLearner benefits the most, achieving the lowest MSE in nearly all settings. Its adaptive weighting mechanism proves particularly effective under high-noise and small-sample scenarios, where fixed-weight methods like XLearner tend to suffer. Notably, RXLearner maintains strong performance even in higher dimensions, highlighting its robustness and favorable convergence behavior compared to traditional meta-learners.
2. In Simulation 3, under confounded treatment assignment, the TLearner exhibits the highest estimation error, followed by the SLearner. In contrast, the XLearner and RXLearner maintain robust performance.
3. In Simulation 4, where the treatment effect is null, the SLearner performs the best, consistent with its underlying model assumptions. Furthermore, our quantitative evaluation of the false positive rate (FPR) under this setting reveals that the SLearner maintains the lowest FPR across varying thresholds and sample sizes, followed by RXLearner, while XLearner and TLearner tend to produce spurious heterogeneity more frequently. This underscores the importance of cautious model selection when the true effect is absent.
4. In Simulation 5, which involves extremely unbalanced treatment assignment, the RXLearner successfully inherits the strength of the XLearner, while the TLearner performs the worst due to its separate modeling for each treatment group.

5. In Simulation 6, the four learners exhibit similar performance across different levels of variable correlation. The estimation accuracy remains stable under noise levels $\sigma = 0.5$ and $\sigma = 1$, with only a slight deterioration observed when $\sigma = 2$.

In summary, the RXlearner consistently demonstrates the best performance across various scenarios. By enhancing the weighting mechanism of the Xlearner, RXlearner maintains equal or superior accuracy in nearly all settings.

5. Applications

To illustrate our proposed method, we analyze a large-scale Get-Out-the-Vote (GOTV) experiment, which is the same experiment used by [10] to test the Xlearner. This experiment investigates whether social pressure can be used to increase voter turnout in U.S. elections. The authors considered all voters who participated in the 2004 general election as registered voters, randomly selected a subset, and assigned them to either the treatment or control group. Households in the treatment group were sent a mailer with the message “DO YOUR CIVIC DUTY—VOTE!”, and the outcomes were observed during the 2006 primary election. We follow the method of [10] for simulation, but differ in the selection of covariates. While social pressure is typically transmitted through nearby neighbors, it is also influenced by the number of household members. Therefore, we include eight covariates, adding the number of household members as the eighth covariate, in contrast to [10], who used gender, age, and voting history in the 2000, 2002, and 2004 primary elections and the 2000 and 2002 general elections.

A common challenge in evaluating the accuracy of heterogeneous treatment effect estimators on real data is the lack of ground truth. To address this, we introduce synthetic treatment effects into the original dataset. We use the CATE estimates generated by the random forest-based Slearner, Tlearner, Xlearner and RXlearner as the ground truth. This allows us to assign potential outcomes to each sample and create a complete dataset. We can then verify whether different methods successfully recover the true effects and investigate whether the CATE estimates from different estimators significantly impact the results. We select 1000 and 10,000 samples from the full dataset as training sets, with the remaining data used as test sets. The proportion of treated and control groups in the selected samples matches the full dataset with $\mathbb{P}(W = 1) = 0.167$. Figures 8 and 9 present the results of this experiment. We find that the CATE estimates from different methods have a relatively minor impact on the overall model performance. However, in smaller samples ($n = 1000$), the Slearner outperforms the Tlearner, while the opposite is true for larger samples ($n = 10,000$).

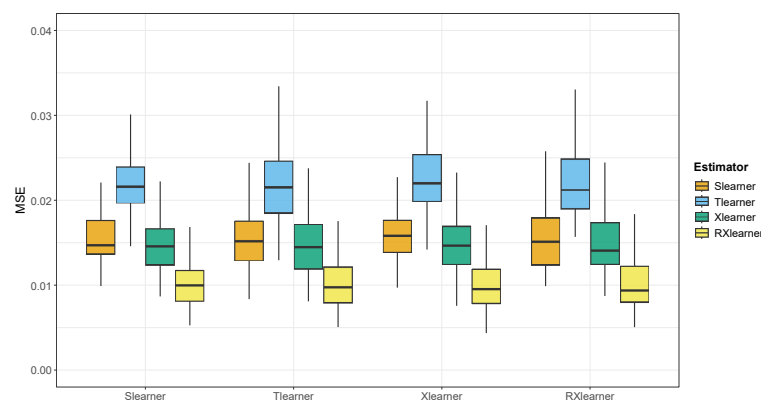


Figure 8. The boxplots of MSE of estimated CATE for each method when the training sample size is $n = 1000$, based on synthetic data generated using the fitted CATE from Slearner, Tlearner, Xlearner and RXlearner, respectively.

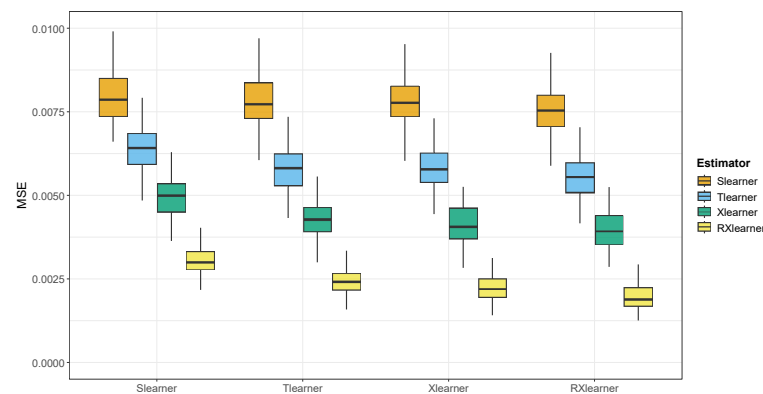


Figure 9. The boxplots of MSE for each method when the training sample size is $n = 10,000$, based on synthetic data generated using the fitted CATE from Slearner, Tlearner, Xlearner and RXlearner, respectively.

Notably, we observe that RXlearner achieves a significantly lower MSE compared to other methods when $n = 1000$. For the other meta-learners, the performance trends are consistent with those reported in [10], where the performance gaps between methods become more pronounced in low-sample regimes. We believe that in this case, there is still room for improvement in how Xlearner selects the weighting function $g(x)$, and the optimization of $g(x)$ in RXlearner leads to a more effective combination of pseudo-responses, resulting in superior performance.

Overall, the RXlearner demonstrates significantly better performance compared to the other estimators.

6. Conclusions

This paper reviews CATE estimation methods under the meta-learner framework, including the Slearner, Tlearner, and Xlearner. Additionally, we propose a new algorithm, the RXlearner. The RXlearner retains the advantages of the Xlearner in handling unbalanced data while being more robust and effective. We conduct extensive simulation studies and real-data experiments, demonstrating that the RXlearner performs excellently in most scenarios. The error bounds of the estimator vary depending on the continuity of the response function. We establish error bounds for the case where the response function is Hölder continuous and show that using KNN as the base learner can achieve these bounds.

There are still many potential improvements for the RXlearner, such as incorporating ideas from DML by applying cross-fitting to (8) [25], or adopting a more direct approach by adding structure to (8), as explored in the works of [26,27]. Another promising direction is to incorporate regularization or complexity control into the second-stage optimization in order to further mitigate the risk of overfitting. We leave the extension of the RXlearner to future work.

Author Contributions: Methodology, Z.Z.; Software, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Youth Academic Innovation Team Construction project of Capital University of Economics and Business, grant number QNTD202303.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1

Proof of Theorem 1. Many ideas in this proof are inspired by [10]; however, our analysis is conducted under the weaker assumption of Hölder continuity. We proceed by contradiction.

Let $a = 2\alpha / (2\alpha + d)$, and suppose $a_2 \geq a$. Denote by C_H the class of Hölder continuous functions $f : [0, 1]^d \rightarrow \mathcal{R}$. For any $f_1 \in C_H$, define $\mathcal{D}(f_1) \in \mathcal{D}_{mn}^H$ as the joint distribution where $\mu_0 = 0$, $\mu_1 = f_1$, and $\varepsilon_0 \perp \varepsilon_1$. Then, Equation (8) implies the following:

$$\begin{aligned} c(m^{-a_1} + n^{-a_2}) &\geq \sup_{\mathcal{P} \in \mathcal{D}_{mn}^H} \mathbb{E}_{(\mathcal{D}_0^m \times \mathcal{D}_1^n) \sim \mathcal{P}} \left[\left(\tau^{\mathcal{P}}(\mathcal{X}) - \hat{\tau}^{mn}(\mathcal{X}; \mathcal{D}_0^m, \mathcal{D}_1^n) \right)^2 \right] \\ &\geq \sup_{f_1 \in C_H} \mathbb{E}_{(\mathcal{D}_0^m \times \mathcal{D}_1^n) \sim \mathcal{D}(f_1)} \left[\left(\mu_1^{\mathcal{D}(f_1)}(\mathcal{X}) - \hat{\tau}^{mn}(\mathcal{X}; \mathcal{D}_0^m, \mathcal{D}_1^n) \right)^2 \right]. \end{aligned}$$

Let $\mathcal{P}_0$ denote the marginal distribution of the control group $\mathcal{D}_0^m = \{(X_i^0, Y_i^0)\}_{i=1}^m$ under $\mathcal{D}(f_1)$. We then have

$$\begin{aligned} c(m^{-a_1} + n^{-a_2}) &\geq \sup_{f_1 \in C_H} \mathbb{E}_{\mathcal{D}_1^n \sim \mathcal{D}_1(f_1)} \mathbb{E}_{\mathcal{D}_0^m \sim \mathcal{P}_0} \left[\left(\mu_1^{\mathcal{D}(f_1)}(\mathcal{X}) - \hat{\tau}^{mn}(\mathcal{X}; \mathcal{D}_0^m, \mathcal{D}_1^n) \right)^2 \right] \\ &\geq \sup_{f_1 \in C_H} \mathbb{E}_{\mathcal{D}_1^n \sim \mathcal{D}_1(f_1)} \left[\left(\mu_1^{\mathcal{D}(f_1)}(\mathcal{X}) - \mathbb{E}_{\mathcal{D}_0^m \sim \mathcal{P}_0} \hat{\tau}^{mn}(\mathcal{X}; \mathcal{D}_0^m, \mathcal{D}_1^n) \right)^2 \right], \end{aligned}$$

where the last inequality follows from Jensen's inequality.

Now, let m_n be a sequence such that $m_n^{-a_1} + n^{-a_2} \leq 2n^{-a_2}$. Define

$$\hat{\mu}_1^n(x; \mathcal{D}_1^n) := \mathbb{E}_{\mathcal{D}_0^{m_n} \sim \mathcal{P}_0^{m_n}} [\hat{\tau}^{m_n}(\mathcal{X}; \mathcal{D}_0^{m_n}, \mathcal{D}_1^n)],$$

and observe that

$$\{\mathcal{D}_1(f_1) : f_1 \in C_H\} = \{\mathcal{P}_1 \in \mathcal{F}^H\}.$$

Hence, we can derive

$$\begin{aligned} 2cn^{-a_2} &\geq c(m_n^{-a_1} + n^{-a_2}) \\ &\geq \sup_{f_1 \in C_H} \mathbb{E}_{\mathcal{D}_1^n \sim \mathcal{D}_1(f_1)} \left[\left(\mu_1^{\mathcal{D}(f_1)}(\mathcal{X}) - \hat{\mu}_1^n(\mathcal{X}; \mathcal{D}_1^n) \right)^2 \right] \\ &\geq \sup_{\mathcal{P}_1 \in \mathcal{F}^H} \mathbb{E}_{\mathcal{D}_1^n \sim \mathcal{P}_1^n} \left[\left(\mu_1^{\mathcal{P}_1}(\mathcal{X}) - \hat{\mu}_1^n(\mathcal{X}; \mathcal{D}_1^n) \right)^2 \right], \end{aligned}$$

which contradicts the minimax lower bound for nonparametric regression under Hölder continuity when $a_2 > a = 2\alpha / (2\alpha + d)$. This completes the proof. $\square$

Appendix A.2

To prove Theorem 2, we first introduce the following two auxiliary results.

Lemma A1. Let $x \in [0, 1]^d$, and suppose $X_1, \dots, X_n \stackrel{i.i.d.}{\sim} \text{Unif}([0, 1]^d)$ with $d > 2$. Let $\tilde{X}(x)$ denote the nearest neighbor of x among $\{X_1, \dots, X_n\}$. Then there exists a constant $c > 0$, independent of n , such that

$$\mathbb{E}[\|\tilde{X}(x) - x\|^{2\alpha}] \leq \frac{c}{n^{2\alpha/d}}.$$

Proof of Lemma A1. Observe that

$$\mathbb{P}(\|\tilde{X}(x) - x\| \geq \delta) = (1 - \mathbb{P}(\|X_1 - x\| \leq \delta))^n \leq (1 - \bar{c}\delta^d)^n \leq e^{-\bar{c}\delta^d n},$$

for some constant $\tilde{c} > 0$. Hence,

$$\begin{aligned}\mathbb{E}\left[\|\tilde{X}(x) - x\|^{2\alpha}\right] &= \int_0^\infty \mathbb{P}\left(\|\tilde{X}(x) - x\|^{2\alpha} \geq \delta\right) d\delta = \int_0^\infty \mathbb{P}\left(\|\tilde{X}(x) - x\| \geq \delta^{1/(2\alpha)}\right) d\delta \\ &\leq \int_0^\infty \exp\left(-\tilde{c}\delta^{d/(2\alpha)}n\right) d\delta = cn^{-2\alpha/d},\end{aligned}$$

for some constant $c > 0$, which completes the proof. $\square$

Lemma A2. Let $\hat{\mu}_0^m$ be the k -nearest neighbors (kNN) estimator with fixed k_0 , computed using only the control group, and let $\hat{\mu}_1^n$ be the kNN estimator with fixed k_1 , computed using only the treated group. Under the assumptions in Theorem 2, we have

$$\mathbb{E}[\|\hat{\mu}_1^n - \mu_1\|^2] \leq \frac{\sigma^2}{k_1} + cL^2\left(\frac{k_1}{n}\right)^{2\alpha/d} \quad \text{and} \quad \mathbb{E}[\|\hat{\mu}_0^m - \mu_0\|^2] \leq \frac{\sigma^2}{k_0} + cL^2\left(\frac{k_0}{m}\right)^{2\alpha/d}.$$

Proof of Lemma A2. We provide the proof for the first bound; the argument for $\hat{\mu}_0^m$ is analogous.

We decompose the mean squared error as follows

$$\begin{aligned}\mathbb{E}\left[(\hat{\mu}_1^n(x) - \mu_1(x))^2\right] &= \mathbb{E}\left[(\hat{\mu}_1^n(x) - \mathbb{E}[\hat{\mu}_1^n(x) \mid X_1, \dots, X_n])^2\right] \\ &\quad + \mathbb{E}\left[(\mathbb{E}[\hat{\mu}_1^n(x) \mid X_1, \dots, X_n] - \mu_1(x))^2\right] \\ &= I_1 + I_2.\end{aligned}$$

For the variance term I_1 , we have

$$\begin{aligned}I_1 &= \mathbb{E}\left[\left(\frac{1}{k_1} \sum_{i=1}^{k_1} (Y_{(i,n)}(x) - \mu_1(X_{(i,n)}(x)))\right)^2\right] \\ &= \frac{1}{k_1^2} \sum_{i=1}^{k_1} \mathbb{E}\left[\text{Var}(Y_{(i,n)}(x) \mid X_{(i,n)}(x))\right] \\ &= \frac{1}{k_1^2} \sum_{i=1}^{k_1} \mathbb{E}\left[\sigma^2(X_{(i,n)}(x))\right] \leq \frac{\sigma^2}{k_1}.\end{aligned}$$

For the squared bias term I_2 , using Hölder continuity of $\mu_1(\cdot)$, we have

$$\begin{aligned}I_2 &= \mathbb{E}\left[\left(\frac{1}{k_1} \sum_{i=1}^{k_1} (\mu_1(X_{(i,n)}(x)) - \mu_1(x))\right)^2\right] \\ &\leq \mathbb{E}\left[\left(\frac{1}{k_1} \sum_{i=1}^{k_1} C\|X_{(i,n)}(x) - x\|^\alpha\right)^2\right] \\ &\leq C^2 \mathbb{E}\left[\left(\frac{1}{k_1} \sum_{i=1}^{k_1} \|X_{(i,n)}(x) - x\|^\alpha\right)^2\right].\end{aligned}$$

To bound this expression, let $N = k_1 \lfloor n/k_1 \rfloor$, and divide the dataset into $k_1 + 1$ disjoint subsets. Define $\tilde{X}_j^x$ to be the nearest neighbor to x in the j -th subset. Then, $\tilde{X}_1^x, \dots, \tilde{X}_{k_1}^x$ are independent and satisfy

$$\sum_{i=1}^{k_1} \|X_{(i,n)}(x) - x\|^\alpha \leq \sum_{j=1}^{k_1} \|\tilde{X}_j^x - x\|^\alpha.$$

By Jensen's inequality

$$\begin{aligned} I_2(x) &\leq C^2 \mathbb{E} \left[\left(\frac{1}{k_1} \sum_{j=1}^{k_1} \|\tilde{X}_j^x - x\|^\alpha \right)^2 \right] \\ &\leq C^2 \frac{1}{k_1} \sum_{j=1}^{k_1} \mathbb{E} [\|\tilde{X}_j^x - x\|^{2\alpha}] \\ &= C^2 \mathbb{E} [\|X_{(1, \lfloor n/k_1 \rfloor)}(x) - x\|^{2\alpha}]. \end{aligned}$$

Integrating over the distribution of x (with $p(x)$ as its density function) and applying Lemma A1, we obtain:

$$\begin{aligned} \frac{1}{C^2} \left[\frac{n}{k_1} \right]^{2\alpha/d} \int I_2(x) p(x) dx &\leq \left[\frac{n}{k_1} \right]^{2\alpha/d} \mathbb{E} [\|X_{(1, \lfloor n/k_1 \rfloor)}(X) - X\|^{2\alpha}] \\ &\leq \text{const.} \end{aligned}$$

Combining the bounds for I_1 and I_2 completes the proof. $\square$

Proof of Theorem 2. We begin by decomposing the RXlearner estimator $\hat{\tau}_1^{mn}(x)$ as

$$\hat{\tau}_1^{mn}(x) = \frac{1}{k_1} \sum_{i=1}^{k_1} [Y_{(i,n)}^1(x) - \hat{\mu}_0^m(X_{(i,n)}^1(x))] = \hat{\mu}_1^n(x) - \frac{1}{k_1} \sum_{i=1}^{k_1} \hat{\mu}_0^m(X_{(i,n)}^1(x)),$$

where the stage-one estimators are defined as

$$\hat{\mu}_0^m(x) = \frac{1}{k_0} \sum_{j=1}^{k_0} Y_{(j,m)}^0(x), \quad \hat{\mu}_1^n(x) = \frac{1}{k_1} \sum_{i=1}^{k_1} Y_{(i,n)}^1(x).$$

We evaluate the mean squared error as follows

$$\begin{aligned} \mathbb{E} [|\tau(\mathcal{X}) - \hat{\tau}_1^{mn}(\mathcal{X})|^2] &= \mathbb{E} \left[\left| \mu_1(\mathcal{X}) - \mu_0(\mathcal{X}) - \hat{\mu}_1^n(\mathcal{X}) + \frac{1}{k_1} \sum_{i=1}^{k_1} \hat{\mu}_0^m(X_{(i,n)}^1(\mathcal{X})) \right|^2 \right] \\ &\leq 2\mathbb{E} [|\mu_1(\mathcal{X}) - \hat{\mu}_1^n(\mathcal{X})|^2] + 2\mathbb{E} \left[\left| \mu_0(\mathcal{X}) - \frac{1}{k_1} \sum_{i=1}^{k_1} \hat{\mu}_0^m(X_{(i,n)}^1(\mathcal{X})) \right|^2 \right] \\ &=: 2I_3 + 2I_4. \end{aligned}$$

From Lemma A2, we have

$$I_3 = \mathbb{E} [\|\hat{\mu}_1^n - \mu_1\|^2] \leq \frac{\sigma^2}{k_1} + c_1 L^2 \left(\frac{k_1}{n} \right)^{2\alpha/d}.$$

To bound I_4 , we decompose it into two terms

$$I_4 \leq \underbrace{\mathbb{E} \left[\left| \mu_0(\mathcal{X}) - \frac{1}{k_1 k_0} \sum_{i=1}^{k_1} \sum_{j=1}^{k_0} \mu_0(X_{(j,m)}^0(X_{(i,n)}^1(\mathcal{X}))) \right|^2 \right]}_{(a)} \quad (\text{A1})$$

$$+ \underbrace{\mathbb{E} \left[\left| \frac{1}{k_1 k_0} \sum_{i=1}^{k_1} \sum_{j=1}^{k_0} \mu_0(X_{(j,m)}^0(X_{(i,n)}^1(\mathcal{X}))) - \frac{1}{k_1} \sum_{i=1}^{k_1} \hat{\mu}_0^m(X_{(i,n)}^1(\mathcal{X})) \right|^2 \right]}_{(b)}. \quad (\text{A2})$$

Term (A2) can be bounded by

$$(b) \leq \max_i \frac{1}{k_0^2} \sum_{j=1}^{k_0} \mathbb{E} \left[\left(\mu_0 \left(X_{(j,m)}^0 \left(X_{(i,n)}^1(\mathcal{X}) \right) \right) - Y_{(j,m)}^0 \left(X_{(i,n)}^1(\mathcal{X}) \right) \right)^2 \right] \leq \frac{\sigma^2}{k_0},$$

using the fact that $Y_{(j,m)}^0(x) \sim \mathcal{N}(\mu_0(x), \sigma^2)$.

Term (A1) is bounded via Hölder continuity and Jensen's inequality

$$\begin{aligned} (a) &\leq \mathbb{E} \left[\left(\frac{1}{k_1 k_0} \sum_{i=1}^{k_1} \sum_{j=1}^{k_0} L \|\mathcal{X} - X_{(j,m)}^0(X_{(i,n)}^1(\mathcal{X}))\|^\alpha \right)^2 \right] \\ &\leq \frac{L^2}{k_1 k_0} \sum_{i=1}^{k_1} \sum_{j=1}^{k_0} \mathbb{E} \left[\|\mathcal{X} - X_{(j,m)}^0(X_{(i,n)}^1(\mathcal{X}))\|^{2\alpha} \right] \\ &\leq L^2 \left(\frac{1}{k_1} \sum_{i=1}^{k_1} \mathbb{E} \|\mathcal{X} - X_{(i,n)}^1(\mathcal{X})\|^{2\alpha} \right. \\ &\quad \left. + \frac{1}{k_1 k_0} \sum_{i=1}^{k_1} \sum_{j=1}^{k_0} \mathbb{E} \|X_{(i,n)}^1(\mathcal{X}) - X_{(j,m)}^0(X_{(i,n)}^1(\mathcal{X}))\|^{2\alpha} \right). \end{aligned} \quad (\text{A3})$$

Applying Lemma A1 to each of the above terms yields

$$(a) \leq 2\tilde{c}L^2 \left(\left(\frac{k_1}{n} \right)^{2\alpha/d} + \left(\frac{k_0}{m} \right)^{2\alpha/d} \right).$$

Putting all pieces together, we obtain

$$\begin{aligned} \mathbb{E} \left[|\tau(\mathcal{X}) - \hat{\tau}_1^{mn}(\mathcal{X})|^2 \right] &\leq 2\frac{\sigma^2}{k_1} + 2(c_1 + 4\tilde{c})L^2 \left(\frac{k_1}{n} \right)^{2\alpha/d} + 4\frac{\sigma^2}{k_0} + 8\tilde{c}L^2 \left(\frac{k_0}{m} \right)^{2\alpha/d} \\ &\leq C \left(\frac{\sigma^2}{k_1} + L^2 \left(\frac{k_1}{n} \right)^{2\alpha/d} + \frac{\sigma^2}{k_0} + L^2 \left(\frac{k_0}{m} \right)^{2\alpha/d} \right), \end{aligned}$$

for a constant $C = 2 \max\{2, c_1 + 4\tilde{c}, 4\tilde{c}\}$. $\square$

References

- Robins, J.M.; Mark, S.D.; Newey, W.K. Estimating Exposure Effects by Modelling the Expectation of Exposure Conditional on Confounders. *Biometrics* **1992**, *48*, 479–495. [\[CrossRef\]](#) [\[PubMed\]](#)
- Robinson, P.M. Root-N-Consistent Semiparametric Regression. *Econometrica* **1988**, *56*, 931–954. [\[CrossRef\]](#)
- Ravikumar, P.; Lafferty, J.; Liu, H.; Wasserman, L. Sparse Additive Models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2009**, *71*, 1009–1030. [\[CrossRef\]](#)
- Robins, J.; Li, L.; Tchetgen, E.; Vaart, A. Higher Order Influence Functions and Minimax Estimation of Nonlinear Functionals. *J. Am. Stat. Assoc.* **2008**, *2*, 335–421. [\[CrossRef\]](#)
- Horvitz, D.G.; Thompson, D.J. A Generalization of Sampling Without Replacement from a Finite Universe. *J. Am. Stat. Assoc.* **1952**, *47*, 663–685. [\[CrossRef\]](#)
- Hirano, K.; Imbens, G.W.; Ridder, G. Efficient Estimation of Average Treatment Effects Using the Estimated Propensity Score. *Econometrica* **2003**, *71*, 1161–1189. [\[CrossRef\]](#)
- Robins, J.M.; Rotnitzky, A. Semiparametric Efficiency in Multivariate Regression Models with Missing Data. *J. Am. Stat. Assoc.* **1995**, *90*, 122–129. [\[CrossRef\]](#)
- Imai, K.; Ratkovic, M. Covariate Balancing Propensity Score. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2014**, *76*, 243–263. [\[CrossRef\]](#)
- Van der Laan, M.J. Statistical Inference for Variable Importance. *Int. J. Biostat.* **2006**, *2*, 1–20. [\[CrossRef\]](#)
- Künzel, S.R.; Sekhon, J.S.; Bickel, P.J.; Yu, B. Metalearners for Estimating Heterogeneous Treatment Effects Using Machine Learning. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 4156–4165. [\[CrossRef\]](#)
- Athey, S.; Imbens, G.W. Machine Learning Methods for Estimating Heterogeneous Causal Effects. *Stat* **2015**, *1050*, 1–26.

12. Athey, S.; Imbens, G. Recursive Partitioning for Heterogeneous Causal Effects. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 7353–7360. [[CrossRef](#)]
13. Nie, X.; Wager, S. Quasi-Oracle Estimation of Heterogeneous Treatment Effects. *Biometrika* **2021**, *108*, 299–319. [[CrossRef](#)]
14. Athey, S.; Tibshirani, J.; Wager, S. Generalized Random Forests. *Ann. Stat.* **2019**, *47*, 1148–1178. [[CrossRef](#)]
15. Chernozhukov, V.; Chetverikov, D.; Demirer, M.; Duflo, E.; Hansen, C.; Newey, W.; Robins, J. Double/Debiased Machine Learning for Treatment and Structural Parameters. *Econom. J.* **2018**, *21*, C1–C68. [[CrossRef](#)]
16. Kennedy, E.H. Towards Optimal Doubly Robust Estimation of Heterogeneous Causal Effects. *Electron. J. Stat.* **2023**, *17*, 3008–3049. [[CrossRef](#)]
17. Li, R.; Wang, H.; Tu, W. Robust Estimation of Heterogeneous Treatment Effects Using Electronic Health Record Data. *Stat. Med.* **2021**, *40*, 2713–2752. [[CrossRef](#)] [[PubMed](#)]
18. Neyman, J. Sur les Applications de la Théorie des Probabilités aux Expériences Agricoles: Essai des Principes. *Rocz. Nauk Rol.* **1923**, *10*, 1–51.
19. Rubin, D.B. Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies. *J. Educ. Psychol.* **1974**, *66*, 688. [[CrossRef](#)]
20. Cox, D.R. The Interpretation of the Effects of Non-Additivity in the Latin Square. *Biometrika* **1958**, *45*, 69–73. [[CrossRef](#)]
21. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
22. Györfi, L.; Kohler, M.; Krzyzak, A.; Walk, H. *A Distribution-Free Theory of Nonparametric Regression*; Springer Science & Business Media: New York, NY, USA, 2006.
23. White, C.R.; Seymour, R.S. Mammalian Basal Metabolic Rate is Proportional to Body Mass^{2/3}. *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 4046–4049. [[CrossRef](#)] [[PubMed](#)]
24. Friedman, J.H. Multivariate Adaptive Regression Splines. *Ann. Stat.* **1991**, *19*, 1–67. [[CrossRef](#)]
25. Schick, A. On Asymptotically Efficient Estimation in Semiparametric Models. *Ann. Stat.* **1986**, *14*, 1139–1151. [[CrossRef](#)]
26. Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *J. R. Stat. Soc. Ser. B (Methodol.)* **1996**, *58*, 267–288. [[CrossRef](#)]
27. Imai, K.; Ratkovic, M. Estimating Treatment Effect Heterogeneity in Randomized Program Evaluation. *Ann. Appl. Stat.* **2013**, *7*, 443–470. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.