



Transfer learning for causal forest

Bérénice-Alexia Jocteur, Véronique Maume-Deschamps, Pierre Ribereau

► To cite this version:

Bérénice-Alexia Jocteur, Véronique Maume-Deschamps, Pierre Ribereau. Transfer learning for causal forest. 2025. [hal-04700817v2](https://hal.science/hal-04700817v2)

HAL Id: hal-04700817

<https://hal.science/hal-04700817v2>

Preprint submitted on 30 Mar 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC-ND 4.0 - Attribution - Non-commercial use - No Derivative Works - International License

Transfer learning for causal forests

Bérénice-Alexia Jocteur, Véronique Maume-Deschamps,
Pierre Ribereau

Université Lyon 1, Centrale Lyon, INSA Lyon,
Université Jean Monnet, CNRS, ICJ UMR5208, 69622 Villeurbanne, France.

March 30, 2026

Abstract

Transfer learning addresses the challenge of transferring knowledge from one domain to another. Traditional transfer learning focuses on adapting models trained on a source domain (with many observations) to improve performance on a target domain (with few observations). In this work, we consider the case of a model shift and focus on transfer learning applied to a causal forest, namely HTERF. This causal forest aims to estimate the Conditional Average Treatment Effect (CATE). The approach considered is the offset method presented by Wang (2016), adapted to a causal context. This method relies on the use of intermediate models to estimate the offset between source and target distributions. We establish an L^1 -consistency result of our algorithm and derive a bound on the CATE estimation error of HTERF in the target domain, depending on the error of the intermediate models. Simulation studies demonstrate the good performance of this approach in different settings.

keywords: causal inference, causal forest, transfer learning, domain adaptation, offset method

1 Introduction

Estimating causal effects is essential for answering *what-if* questions in many application fields, such as policy evaluation, medicine, or business strategy. The *potential outcomes* framework Imbens and Rubin (2015) defines and estimates causal quantities such as the *Average Treatment Effect* (ATE) and the *Conditional Average Treatment Effect* (CATE). However, a major challenge arises when the target domain—where predictions are needed—has limited labeled data, while a related source domain has abundant observations. This scenario is common in real-world applications, where data distribution shifts, such as *model shift* (i.e., $P(Y^t(w)|\mathbf{X}^t) \neq P(Y^s(w)|\mathbf{X}^s)$)

for $w \in \{0, 1\}$), can undermine the performance of traditional causal inference methods.

In this context, *transfer learning* enables knowledge transfer from a data-rich source domain to a data-scarce target domain. Among transfer learning strategies, the *offset method* Wang (2016), originally designed for regression tasks, adjusts the source domain predictions using a correction learned from the target domain. This approach is particularly appealing in causal inference, where the goal is to estimate the CATE in the target domain while leveraging the structural information available in the source domain.

In this work, we focus on adapting the offset method to the causal setting, specifically for estimating the CATE under model shift. Our approach integrates the offset method with the *Heterogeneous Treatment Effect-based Random Forest* (HTERF) Jocteur et al. (2024), a specialized random forest algorithm designed for CATE estimation. HTERF improves upon existing methods, such as Generalized Random Forests Athey et al. (2019), by employing a splitting criterion tailored to capture treatment effect heterogeneity. We propose to combine the offset method with HTERF.

Our contributions are threefold:

- We introduce a causal adaptation of the offset method, with two variants: one that separately models treated and control groups, and another that considers the treatment indicator as an additional covariate.
- We establish an L^1 -consistency result for the CATE estimator in the target domain.
- We derive a generalization bound that highlights the role of intermediate model errors in the transfer process.

The paper is organized as follows. Section 2 contains the necessary background on causality, the HTERF algorithm, and domain adaptation. In Section 3, we detail the offset method. Section 4 presents our adaptation of the offset method to CATE estimation and our L^1 consistency result, along with a generalization bound. A simulation study is presented in Section 5. A short discussion can be found in Section 6. Most technical proofs are postponed to the final Appendix.

2 Some background

In this section, we provide a brief overview of the background necessary to understand the article. It includes summary elements on the causal framework, the HTERF algorithm, and domain adaptation.

2.1 The causal framework

Following the framework outlined in Imbens and Rubin (2015), the potential outcomes denoted $Y(1)$ and $Y(0)$ are defined as the outcomes that would have been observed if treatment or control had been assigned to the quantity of interest Y , respectively. Let $Y = Y(W)$ be the observed outcome, where W represents a binary treatment. Additionally, we incorporate a set of covariates $\mathbf{X} \in \mathbb{R}^d$. The conditional average treatment effect (CATE) at \mathbf{x} is defined as follows:

$$\tau(\mathbf{x}) = \mathbb{E}[Y(1) - Y(0) | \mathbf{X} = \mathbf{x}]. \quad (2.1)$$

The average treatment effect (ATE) is:

$$\mathbb{E}[Y(1) - Y(0)].$$

A standard assumption for the identifiability of CATE is unconfoundedness (Rosenbaum and Rubin (1983)), meaning that conditionally on \mathbf{X} , the treatment assignment W is independent of the potential outcomes for Y :

$$\{Y(1), Y(0)\} \perp\!\!\!\perp W | \mathbf{X}. \quad (2.2)$$

Many algorithms in the literature allow for the evaluation of CATE: causal forests, meta-learners, and causal neural networks, among others. In what follows, we focus on HTERF, a short presentation of which is given in the next section.

2.2 HTERF

The transfer algorithms we present are based on HTERF, a special case of random forest introduced in Jocteur et al. (2024). It differs from the GRF model of Athey et al. (2019) by a new splitting criterion used to construct the trees. Given a sample $(W_i, \mathbf{X}_i, Y_i)_{i=1, \dots, n} \in \{0, 1\} \times \mathbb{R}^d \times \mathbb{R}$, it provides an estimator $\hat{\tau}_{B,n}(\mathbf{x})$ of the CATE $\tau(\mathbf{x})$. With some assumptions on the distribution of (W, \mathbf{X}, Y) and on the construction of the forest, an almost sure convergence result of $\hat{\tau}_{B,n}(\cdot)$ to $\tau(\cdot)$ is obtained, as well as an interpretability result. This algorithm has been implemented in `Julia`, in the package `CausalForest`, Jocteur (2023)¹. For completeness, we summarize below the main points in HTERF construction. For further background knowledge on the HTERF algorithm and the associated theoretical results, we refer the readers to Jocteur et al. (2024). Let us remark that HTERF outperforms GRF and meta-learners on most of the tested settings.

The splitting criterion used in HTERF is designed to maximize the difference in treatment effect between child nodes. Specifically, Definition (2.1)

¹<https://juliapackages.com/p/causalforest>

is used to define this splitting criterion. The tree-building process of HTERF relies on the optimization of the splitting criterion $\Delta(A, j, z)$:

$$\Delta(A, j, z) = \frac{|A_L||A_R|}{|A|^2} \left((\bar{Y}_{A_{L1}} - \bar{Y}_{A_{L0}}) - (\bar{Y}_{A_{R1}} - \bar{Y}_{A_{R0}}) \right)^2,$$

where A is an hyper-rectangle of the form $\prod_{i=1}^d [a_i, b_i]$, j belongs to a randomly chosen subset \mathcal{M}_{try} of $\{1, \dots, d\}$, $z \in A^j = [a_j, b_j]$, \bar{Y}_A is the empirical mean of the Y_i 's for $\mathbf{X}_i \in A$, and

$$\begin{aligned} A_{L1} &= \{\mathbf{X}_i \in A \mid X_i^{(j)} < z, W_i = 1\}, \\ A_{L0} &= \{\mathbf{X}_i \in A \mid X_i^{(j)} < z, W_i = 0\}, \\ A_{R1} &= \{\mathbf{X}_i \in A \mid X_i^{(j)} \geq z, W_i = 1\}, \\ A_{R0} &= \{\mathbf{X}_i \in A \mid X_i^{(j)} \geq z, W_i = 0\}, \\ A_L &= A_{L1} \cup A_{L0}, \\ A_R &= A_{R1} \cup A_{R0}. \end{aligned}$$

The estimation of τ is then given by:

$$\hat{\tau}_{B,n}(\mathbf{x}) = \sum_{i:W_i=1} \alpha_i(\mathbf{x})Y_i - \sum_{i:W_i=0} \alpha'_i(\mathbf{x})Y_i,$$

where α (respectively, α') are the weights from the forest associated with observations where $W_i = 1$ (respectively, $W_i = 0$).

Let us now give a short overview of transfer learning, which is useful to understand the concepts, even if we shall adapt it to our causal framework context.

2.3 Transfer learning

Transfer learning is a machine learning technique that leverages knowledge gained from solving one problem and applies it to a different but related problem. In traditional machine learning approaches, models are trained from scratch for each task, requiring substantial amounts of labeled data and computational resources. However, in real-world scenarios, labeled data might be scarce or expensive to acquire, hindering the effectiveness of such methods.

Transfer learning addresses these limitations by transferring knowledge from a source domain where labeled data is abundant to a target domain where labeled data is scarce. This approach allows models to generalize better and achieve improved performance, particularly in situations where limited labeled data is available for training.

Domain adaptation is a special case of transfer learning. In domain adaptation, the source and target domains share the same feature space (but have different distributions), while transfer learning includes cases where the target domain’s feature space differs from the source feature space. In what follows, we consider the problem of supervised domain adaptation, where both source and target datasets are labeled.

According to Huyen (2022), in a supervised machine learning problem, the training dataset can be viewed as a set of samples from a joint distribution of $P(\mathbf{X}, Y)$, where \mathbf{X} is the input and Y is the output. We are interested in modeling $P(Y|\mathbf{X})$. Of course, $P(\mathbf{X}, Y)$ can be decomposed as $P(\mathbf{X}|Y) \times P(Y)$ or $P(Y|\mathbf{X}) \times P(\mathbf{X})$.

Different problems are addressed in transfer learning. The most common is the covariate shift, where the marginal distribution $P(\mathbf{X})$ differs between source and target domains, but the conditional distribution $P(Y|\mathbf{X})$ remains the same across the domains. Similarly, label shift can be defined as the case where $P(Y)$ differs between source and target domains, but the conditional distribution $P(\mathbf{X}|Y)$ stays the same across the domains. Finally, model shift or concept drift concerns cases where $P(Y|\mathbf{X})$ changes, but $P(\mathbf{X})$ remains the same.

Different strategies are presented in Huyen (2022) to address these data distribution shifts. The first and simplest strategy is to train models on large and rich datasets, hoping that points following both source and target distributions will be present in this large dataset. This method requires access to large external datasets likely to contain both source and target distributions. Furthermore, it can be costly to train models on very large datasets. A second approach is to use algorithms dedicated to accounting for a certain type of shift; for example, the kernel mean matching (KMM) method (Huang et al. (2006), Gretton et al. (2006)) allows dealing with covariate shift. Zhang et al. (2013) proposes an approach to correct both covariate shift and label shift without using labels from the target distribution (unsupervised domain adaptation problem); similarly, Zhao et al. (2019) proposed domain-invariant representation learning. Wang et al. (2014) introduces two methods to deal with covariate shift in real regression cases, using labeled source data. Finally, a third kind of approach to dealing with data distribution shift is to retrain the model with labeled target data, either

by retraining the model from scratch with both source and target data or by resuming the training of the existing model (trained on the source) on the target data. This second option, named fine-tuning, is easily applicable to neural networks by using techniques such as freezing layers or warm starting.

Transfer strategies can be extended to the causal context. We consider the source domain (\mathbf{X}^s, Y^s, W^s) and the target domain (\mathbf{X}^t, Y^t, W^t) . We focus on the model shift case, i.e., $P(Y^t(1)|\mathbf{X}^t) \neq P(Y^s(1)|\mathbf{X}^s)$ and $P(Y^t(0)|\mathbf{X}^t) \neq P(Y^s(0)|\mathbf{X}^s)$. We assume that the distributions for \mathbf{X}^s and \mathbf{X}^t (respectively W^s and W^t) are the same. If this were not the case, the distributions of \mathbf{X}^s and \mathbf{X}^t could be matched by various methods dealing with covariate shift (e.g., KMM) without the use of Y . The goal is then to estimate the CATE function on the target population. Recent work has been done to estimate ATE in a supervised domain adaptation setup in Wei et al. (2024), where the nuisance parameters (such as the propensity score) are estimated using ℓ^1 -regularized transfer learning and then plugged into an ATE estimator. We can also mention Künzel et al. (2018), who proposed to transfer knowledge by using several strategies, such as using neural network (NN) weights estimated from the source domain as the warm start of the subsequent target domain NN training, or using NN weights estimated from the source domain and freezing some of its layers before backpropagating through the unfrozen ones when training on the target dataset. Neural networks with an architecture dedicated to causal transfer learning have also been proposed by Bica and van der Schaar (2022). The method we propose is innovative, as it allows transfer learning on CATE estimation without using a neural network.

3 The offset approach

We present the offset approach in a regression context. It is useful to understand the underlying concepts.

3.1 Presentation

Let $\mathcal{X} \in \mathbb{R}^d$ and $\mathcal{Y} \in \mathbb{R}$ be the input and output spaces for a regression task for both source and target domains. Let $(\mathbf{Z}_i^s)_{i \in \{1, \dots, n\}} = ((\mathbf{X}_i^s, Y_i^s))_{i \in \{1, \dots, n\}}$ be the source dataset of size n . We also consider $(\mathbf{Z}_i^{tL})_{i \in \{1, \dots, n_l\}} = ((\mathbf{X}_i^{tL}, y_i^{tL}))_{i \in \{1, \dots, n_l\}}$ as the labeled target dataset of size n_l . There is also an unlabeled target dataset on which we want to test the performance of transfer learning, denoted $(\mathbf{X}_i^{tU})_{i \in \{1, \dots, n_u\}}$ of size n_u .

The offset algorithm (Algorithm 1) introduced by Wang et al. (2014) can be used with any regression machine learning algorithm for each estimator (namely \hat{f}^s, \hat{f}^o, M). In Wang and Schneider (2015), a generalization bound

Algorithm 1 Offset algorithm

Require: A source dataset $\{\mathbf{X}_i^s, Y_i^s\}_{i \in \{1, \dots, n\}}$, a labeled target dataset $\{\mathbf{X}_i^{tL}, Y_i^{tL}\}_{i \in \{1, \dots, n_l\}}$, and an unlabeled target dataset $\{\mathbf{X}_i^{tU}\}_{i \in \{1, \dots, n_u\}}$.
Estimate a model \hat{f}^s that regresses $\{Y_i^s\}$ against $\{\mathbf{X}_i^s\}$.
Estimate a model \hat{f}^o that regresses $\{\hat{Y}_i^o\} = \{Y_i^{tL} - \hat{f}^s(\mathbf{X}_i^{tL})\}$ against $\{\mathbf{X}_i^{tL}\}$.
 $\{Y_i^{new}\} \leftarrow \{Y_i^s + \hat{f}^o(\mathbf{X}_i^s)\}$
Train a model M on $\{\mathbf{X}_i^s, Y_i^{new}\} \cup \{\mathbf{X}_i^{tL}, Y_i^{tL}\}$.
Ensure: $\{\hat{Y}_i^{tU}\} \leftarrow \{M(\mathbf{X}_i^{tU})\}$

is given when Kernel Ridge Regression (KRR) is used in the offset algorithm. For completeness, we provide below a short overview of KRR and the associated generalization bound.

3.2 Using Kernel Ridge Regression

KRR and the related notations are defined as follows.

Definition 1 (Bousquet and Elisseeff (2002)). *Let $\mathcal{S}_T = \{\mathbf{Z}_1 = (\mathbf{X}_1, Y_1), \dots, \mathbf{Z}_n = (\mathbf{X}_n, Y_n)\}$ be a training sample for a regression task in a reproducing kernel Hilbert space (see Wahba (2003)) \mathcal{H} with associated norm $\|\cdot\|_{\mathcal{H}}$. Let ℓ be the l^2 loss function; then the KRR estimator is obtained by: given $\lambda > 0$,*

$$\arg \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(h, \mathbf{Z}_i) + \lambda \|h\|_{\mathcal{H}}^2$$

Two errors are defined:

- $R = \mathbb{E}[\ell(\mathcal{S}_T, \mathbf{Z})]$, the generalization error,
- $R_{emp} = \frac{1}{n} \sum_{i=1}^n \ell(\mathcal{S}_T, \mathbf{Z}_i)$, the empirical error.

The following theorem gives a bound on the generalization error when KRR is used in the offset method.

Theorem 1 (Wang and Schneider (2015)). *If KRR is used to estimate the three functions in the offset method, let R^t be the generalization error on the target dataset of the final model M , R_{emp}^s the empirical error on the source model, and \bar{R}_{emp}^o the empirical error of the estimator \hat{f}^o against $\{\mathbf{X}^{tL}, \hat{Y}^o\}$. Then,*

$$R^t - 2(R_{emp}^s - \bar{R}_{emp}^o) = O\left(\frac{1}{\sqrt{\lambda_o n_l}}\right),$$

where λ_o is the hyperparameter of KRR.

This result relies on Theorem 12 in Bousquet and Elisseeff (2002), which gives a property of uniform stability for the KRR algorithm. However, this property is not known for many other algorithms besides KRR (or only a weaker version of stability is obtained), which makes extensions of this result to the causal case presented in the following section challenging.

4 Causal adaptation

We propose an offset method adapted to the causal framework.

4.1 Overview

Two causal adaptations of the offset method are proposed. In Algorithm 2, the treated and control populations are processed separately to estimate source and offset functions (one estimator for each group). In Algorithm 3, the treatment variable is considered as an additional covariate for the source and offset functions.

The algorithms require a source dataset $\mathcal{D}^s = \{W_i^s, \mathbf{X}_i^s, Y_i^s\}_{i \in \{1, \dots, n\}}$ and a target dataset $\mathcal{D}^t = \{W_i^{tL}, \mathbf{X}_i^{tL}, Y_i^{tL}\}_{i \in \{1, \dots, n_l\}}$.

Algorithm 2 Offset causal algorithm: separate models

Require: A source dataset \mathcal{D}^s , a target dataset \mathcal{D}^t , and an unlabeled target dataset $\{\mathbf{X}_i^{tU}\}_{i \in \{1, \dots, n_u\}}$.

Estimate a model \hat{f}_0^s that regresses $\{Y_i^s\}_{W_i^s=0}$ against $\{\mathbf{X}_i^s\}_{W_i^s=0}$ and a model \hat{f}_1^s that regresses $\{Y_i^s\}_{W_i^s=1}$ against $\{\mathbf{X}_i^s\}_{W_i^s=1}$.

Estimate a model \hat{f}_0^o that regresses $\{Y_i^{tL} - \hat{f}_0^s(\mathbf{X}_i^{tL})\}_{W_i^{tL}=0}$ against $\{\mathbf{X}_i^{tL}\}_{W_i^{tL}=0}$ and a model \hat{f}_1^o that regresses $\{Y_i^{tL} - \hat{f}_1^s(\mathbf{X}_i^{tL})\}_{W_i^{tL}=1}$ against $\{\mathbf{X}_i^{tL}\}_{W_i^{tL}=1}$.

$\{Y_i^{new}\} \leftarrow \{Y_i^s + \hat{f}_{W_i^s}^o(\mathbf{X}_i^s)\}$

Train an HTERF model M on $\{W_i^s, \mathbf{X}_i^s, Y_i^{new}\} \cup \{W_i^{tL}, \mathbf{X}_i^{tL}, Y_i^{tL}\}$.

Ensure: $\{\hat{\tau}^t(\mathbf{X}_i^{tU})\} \leftarrow \{M(\mathbf{X}_i^{tU})\}$

We call $\hat{f}_0^s, \hat{f}_1^s, \hat{f}^s$ the source estimators and $\hat{f}_0^o, \hat{f}_1^o, \hat{f}^o$ the offset estimators. Any regression algorithm could be used to obtain the source and offset estimators. In practice, we obtained good results by using regression random forests.

Algorithm 3 Offset causal algorithm: unique model

Require: A source dataset \mathcal{D}^s , a target dataset \mathcal{D}^t , and an unlabeled target dataset $\{\mathbf{X}_i^{tU}\}_{i \in \{1, \dots, n_u\}}$.

Estimate a model \hat{f}^s that regresses $\{Y_i^s\}$ against $\{\mathbf{X}_i^s, W_i^s\}$.

Estimate a model \hat{f}^o that regresses $\{Y_i^{tL} - \hat{f}^s(\mathbf{X}_i^{tL}, W_i^{tL})\}$ against $\{\mathbf{X}_i^{tL}\}$.

$\{Y_i^{new}\} \leftarrow \{Y_i^s + \hat{f}^o(\mathbf{X}_i^s, W_i^s)\}$

Train an HTERF model M on $\{W_i^s, \mathbf{X}_i^s, Y_i^{new}\} \cup \{W_i^{tL}, \mathbf{X}_i^{tL}, Y_i^{tL}\}$.

Ensure: $\{\hat{\tau}^t(\mathbf{X}_i^{tU})\} \leftarrow \{M(\mathbf{X}_i^{tU})\}$

4.2 Convergence result

For simplicity, in what follows, we shall write indifferently $Y|W$ or $Y(W)$.

Recall that we consider the *model shift* setting, i.e., $\mathbf{X}^s \stackrel{\mathcal{L}}{=} \mathbf{X}^t$, $W^s \stackrel{\mathcal{L}}{=} W^t$, and the conditional laws $Y^s(W^s)|\mathbf{X}^s$ and $Y^t(W^t)|\mathbf{X}^t$ differ.

Since the size of the source dataset is assumed to be large compared to the target dataset, we state a convergence theorem and a generalization bound on the causal offset algorithm if the HTERF model in the last step is only fit on the set $\{W_i^s, \mathbf{X}_i^s, Y_i^{new}; i = 1, \dots, n\} = \mathcal{D}_n$.

We use the following notations, as in Jocteur et al. (2024):

- $\Theta_\ell, \ell = 1, \dots, B$ are independent random vectors, distributed as a generic random vector Θ and independent of \mathcal{D}_n . It contains indices of observations that are used to build each tree; indices of observations that are used for estimations in each tree; indices of splitting candidate variables in each node.
- $\mathcal{D}_{n,1}^*(\Theta_\ell)$ and $\mathcal{D}_{n,2}^*(\Theta_\ell)$ are the disjoint subsamples selected prior to tree construction; the first is used to build the tree, and the second allows the building of weights used during the estimation step.
- $A_n(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)$ is the tree cell (subspace of \mathcal{X}) containing \mathbf{x} .
- $N_{n,1}(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)$ (resp. $N_{n,0}(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)$) is the number of elements of $\mathcal{D}_{n,2}^*(\Theta_\ell)$ that fall into $A_n(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)$, such that $W_i = 1$ (resp. $W_i = 0$).

Let us consider: $\tau_1^t(\mathbf{x}) = \mathbb{E}[Y^t(1)|\mathbf{X}^t = \mathbf{x}]$,

$\tau_0^t(\mathbf{x}) = \mathbb{E}[Y^t(0)|\mathbf{X}^t = \mathbf{x}]$,

$\hat{\tau}_1^{new}(\mathbf{x}) = \sum_{i: W_i=1} \alpha_i(\mathbf{x}) Y_i^{new}$,

and $\hat{\tau}_0^{new}(\mathbf{x}) = \sum_{i: W_i=0} \alpha'_i(\mathbf{x}) Y_i^{new}$, where

$$\alpha_i(\mathbf{x}) = \frac{1}{B} \sum_{\ell=1}^B \frac{\mathbb{1}_{\mathbf{x}_i \in A_n(\mathbf{x}; \Theta_\ell, \mathcal{D}_n) \wedge W_i=1 \wedge i \in \mathcal{D}_{n,2}^*(\Theta_\ell)}}{N_{n,1}(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)}, \quad (4.1)$$

$$\alpha'_i(\mathbf{x}) = \frac{1}{B} \sum_{\ell=1}^B \frac{\mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x}; \Theta_\ell, \mathcal{D}_n) \wedge W_i=0 \wedge i \in \mathcal{D}_{n,2}^*(\Theta_\ell)}}{N_{n,0}(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)}. \quad (4.2)$$

Our aim is to estimate the target CATE:

$$\tau^t(\mathbf{x}) = \tau_1^t(\mathbf{x}) - \tau_0^t(\mathbf{x}).$$

We shall make the following assumptions on the distributions and models. There are slight differences depending on whether Algorithm 2 (two offset models) or Algorithm 3 (one single offset model) is concerned.

Assumption 1.

- \mathbf{X}^s and \mathbf{X}^t are distributed as $\mathbf{X} = (X^{(1)}, \dots, X^{(d)})$, which is a continuous random vector with independent coordinates. The density of \mathbf{X} is positive and bounded from above and below by positive constants.
- \mathbf{X} takes its values in \mathcal{X} , which is assumed to be a positive compact hyper-rectangle of \mathbb{R}^d :

$$\mathcal{X} = \prod_{i=1}^d [u_i, v_i], \quad -\infty < u_i \leq v_i < \infty.$$

- W^s and W^t are distributed as W , a binary variable.

• **Case Algorithm 2**

$Y^s(1) = f_1^s(\mathbf{X}^s) + \varepsilon_1^s$, $\varepsilon_1^s \perp \mathbf{X}^s$,
and $Y^t(1) - f_1^s(\mathbf{X}^t) = f_1^o(\mathbf{X}^t) + \varepsilon_1^t$, $\varepsilon_1^t \perp \mathbf{X}^s, \mathbf{X}^t$.
 ε_1^s and ε_1^t are continuous centered random variables.
 $Y^s(0) = f_0^s(\mathbf{X}^s) + \varepsilon_0^s$, $\varepsilon_0^s \perp \mathbf{X}^s$,
and $Y^t(0) - f_0^s(\mathbf{X}^t) = f_0^o(\mathbf{X}^t) + \varepsilon_0^t$, $\varepsilon_0^t \perp \mathbf{X}^s, \mathbf{X}^t$.
 ε_0^s and ε_0^t are continuous centered random variables.

• **Case Algorithm 3**

$Y^s(W^s) = f^s(\mathbf{X}^s, W^s) + \varepsilon^s$
and $Y^t(W^t) - f^s(\mathbf{X}^t, W^t) = f^o(\mathbf{X}^t, W^t) + \varepsilon^t$.
 ε^s and ε^t are continuous centered random variables, independent respectively of \mathbf{X}^s, W^s and \mathbf{X}^t, W^t .

- $\mathbf{x} \mapsto f^s(\mathbf{x}, W)$, $f_0^s(\mathbf{x})$, $f_1^s(\mathbf{x})$, $f^t(\mathbf{x}, W)$, $f_1^t(\mathbf{x})$, $f_0^t(\mathbf{x})$ are continuous. So in particular $\mathbf{x} \mapsto \tau^t(\mathbf{x})$, $\tau_1^t(\mathbf{x})$, $\tau_0^t(\mathbf{x})$ and $\mathbf{x} \mapsto \tau^s(\mathbf{x})$, $\tau_1^s(\mathbf{x})$, $\tau_0^s(\mathbf{x})$ are continuous.

Assumption 2. The following assumptions are made on B (number of trees in HTERF), $N_{n,1}(\mathbf{x}; \Theta, \mathcal{D}_n)$ resp. $N_{n,0}(\mathbf{x}; \Theta, \mathcal{D}_n)$ (number of observations in a leaf node such as $W = 1$, resp. $W = 0$), and on the construction of the trees:

1. $B = \mathcal{O}(\sqrt{n})$, $\exists C > 0$ and $\exists \beta > 1$ such that $B > C \frac{\sqrt{n}}{(\ln(n))^\beta}$,
2. $\forall \mathbf{x} \in \mathcal{X}$,
 $\mathbb{E}[N_{n,1}(\mathbf{x}; \Theta, \mathcal{D}_n)] = \Omega(\sqrt{n}(\ln(n))^\beta)$.
3. $\forall \mathbf{x} \in \mathcal{X}$,
 $\mathbb{E}[N_{n,0}(\mathbf{x}; \Theta, \mathcal{D}_n)] = \Omega(\sqrt{n}(\ln(n))^\beta)$.
4. $\max_{\mathbf{x}, \Theta} N_{n,1}(\mathbf{x}; \Theta, \mathcal{D}_n) = o(n)$.
5. $\max_{\mathbf{x}, \Theta} N_{n,0}(\mathbf{x}; \Theta, \mathcal{D}_n) = o(n)$.
6. At every step of the tree-building procedure, the probability that the next split is done along the j -th feature is bounded below by π/d for some $0 < \pi \leq 1$ for all $j = 1, \dots, d$.
7. *HTERF*, as described in Jocteur et al. (2024), uses an honest framework: the training sample is split into two parts: one is used to construct the splits of the trees, the other one is used to calculate the weights α and α' . This second subsample verifies that each split leaves at least a fraction δ of the available training sample such that $W = 1$ (resp. $W = 0$) on each side of the split, for some $0 < \delta \leq 0.5$.

Let $\mathcal{D}^s = (W_i^s, \mathbf{X}_i^s, Y_i^s; i = 1, \dots, n)$ be the source sample and $\mathcal{D}^t = (W_i^t, \mathbf{X}_i^t, Y_i^t; i = 1, \dots, n_t)$ be the target sample; $\mathcal{D} = \mathcal{D}^s \cup \mathcal{D}^t$.

Let us first consider the case of Algorithm 2. The function \hat{f}_1^s is an estimator of f_1^s in the first step: if $W^s = 1$,

$$Y^s = \hat{f}_1^s(\mathbf{X}^s) + \varepsilon_1^s + E_1^s(\mathcal{D}^s, \mathbf{X}^s)$$

The function \hat{f}_1^o is an estimator of f_1^o in the second step: if $W^t = 1$,

$$Y^t - \hat{f}_1^s(\mathbf{X}^t) - E_1^s(\mathcal{D}^s, \mathbf{X}^t) = \hat{f}_1^o(\mathbf{X}^t) + E_1^o(\mathcal{D}, \mathbf{X}^t) + \varepsilon_1^t$$

Finally, the third step leads to: if $W^s = 1$,

$$\begin{aligned} Y^{new} &= Y^s + \hat{f}_1^o(\mathbf{X}^s) \\ &= Y^s + Y^t - \hat{f}_1^s(\mathbf{X}^s) - E_1^s(\mathcal{D}^s, \mathbf{X}^s) - \\ &\quad E_1^o(\mathcal{D}, \mathbf{X}^s) - \varepsilon_1^t \\ &= Y^t + \varepsilon_1^s - \varepsilon_1^t - E_1^o(\mathcal{D}, \mathbf{X}^s) \end{aligned}$$

In a similar fashion, we have: if $W^s = 0$,

$$Y^{new} = Y^t + \varepsilon_0^s - \varepsilon_0^t - E_0^o(\mathcal{D}, \mathbf{X}^s)$$

In the case of Algorithm 3, we proceed the same way and are led to:

$$Y^{new}(W^s) = Y^t(W^s) + \varepsilon^s - \varepsilon^t - E^o(\mathcal{D}, \mathbf{X}^s, W^s).$$

These expressions of Y^{new} show that it can be used as an approximation of Y^t distribution.

Then, HTERF is trained on:

$\mathcal{D}_n = \{W_i^s, \mathbf{X}_i^s, Y_i^{new}, i = 1, \dots, n\}$, which gives the following estimator of the target CATE:

$$\begin{aligned} \hat{\tau}_{B,n}^{new}(\mathbf{X}) &= \hat{\tau}_1^{new}(\mathbf{x}) - \hat{\tau}_0^{new}(\mathbf{x}), \\ \text{where } \hat{\tau}_1^{new}(\mathbf{x}) &= \sum_{i:W_i=1} \alpha_i(\mathbf{x}) Y_i^{new} \text{ and} \\ \hat{\tau}_0^{new}(\mathbf{x}) &= \sum_{i:W_i=0} \alpha'_i(\mathbf{x}) Y_i^{new}; \\ \alpha_i(\mathbf{x}) \text{ and } \alpha'_i(\mathbf{x}) &\text{ are defined by Equations (4.1), (4.2).} \end{aligned}$$

Theorem 2. *Let Assumptions 1 and 2 be verified. Assume that for a fixed $\beta > \frac{5}{2}$, $C > 0$, each HTERF tree of the model M is the highest such that $C\sqrt{n}(\ln n)^\beta \leq N_{n,0}(\mathbf{x}; \Theta_\ell, \mathcal{D}_n), N_{n,1}(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)$. Assume that Y^s, Y^t and E^o, E_1^o, E_0^o error terms are bounded a.s. and that E^o, E_1^o, E_0^o converge to 0 in L^2 as n_l tends to ∞ . Then,*

$$\mathbb{E} [|\hat{\tau}_{B,n}^{new}(\mathbf{X}) - \tau^t(\mathbf{X})|] \xrightarrow{n, n_l \rightarrow \infty} 0.$$

The more technical parts of the proof are postponed to Appendix A.

Remark 1. *With estimators \hat{f}_a^s , ($a = 1, 0, \emptyset$) of the form $\sum \omega_i Y_i^s$, since Y is assumed to be bounded, so are \hat{f}_a^s and E_a^s . With \hat{f}_a^o of the form $\sum \omega_i (Y_i^t - \hat{f}_a^s(\mathbf{X}_i^t))$, the error terms E_a^o are also bounded. Most classical regression algorithms provide estimators of this form: random forest, linear regression, neural network, etc.*

Remark 2. *Following what is done in the proof of Theorem 2, the error on the estimation of τ_1^t can be bounded as follows (the same rationale applies for τ_0^t): let $\mathbf{x} \in \mathcal{X}$,*

$$|\hat{\tau}_1^{new}(\mathbf{x}) - \tau_1^t(\mathbf{x})| \leq B_{offset} + B_{HTERF}.$$

Overall, this bound tends to 0 as $n, n_l \rightarrow +\infty$. The second term is the bound of HTERF on a sample of size n , and the first term is introduced by the offset method; the rate of convergence of this quantity only depends on the rate of convergence of \hat{f}^o . More details of these bounds are presented in Appendix B.

Proof of Theorem 2. This proof is partially inspired by the proof of Theorem 4.1 in Jocteur et al. (2024). We provide details in the case of Algorithm 2; the proof for Algorithm 3 follows the same lines by using $\hat{f}^s(\mathbf{X}, W)$ and $\hat{f}^o(\mathbf{X}, W)$ instead of $\hat{f}_1^s(\mathbf{X}), \hat{f}_0^s(\mathbf{X}), \hat{f}_1^o(\mathbf{X}), \hat{f}_0^o(\mathbf{X})$. We emphasize the details induced by the offset method in a causal framework and go more rapidly on parts that could be directly copied from Jocteur et al. (2024).

Let us consider an auxiliary dataset $\mathcal{D}_n^\diamond = (Y_i^\diamond, \mathbf{X}_i^\diamond, W_i^\diamond)_{i=1, \dots, n}$ that is a

sample of (Y^t, \mathbf{X}, W) , independent of \mathcal{D}^t and \mathcal{D}^s . This new sample is used to build $(Y_i^{new, \diamond})_{i=1, \dots, n}$, distributed as $(Y_i^{new})_{i=1, \dots, n}$: if $W_i^\diamond = 1$,

$$Y_i^{new, \diamond} = Y_i^\diamond + \varepsilon_{1,i}^s - \varepsilon_{1,i}^t - E_1^o(\mathcal{D}, \tilde{\mathbf{X}}_i)$$

with $\varepsilon_{1,i}^*$ independent copies of ε_1^* and $\tilde{\mathbf{X}}_i$, $i = 1, \dots, n$ independent copies of \mathbf{X}^s , independent of \mathcal{D}_n^\diamond .

Then, the HTERF trees are grown using \mathcal{D}_n , but the sample \mathcal{D}_n^\diamond (independent of \mathcal{D}_n and Θ) is used to define a dummy estimator

$$\begin{aligned} & \tau^{new, \diamond}(\mathbf{x}; \Theta_1, \dots, \Theta_k, \mathcal{D}_n^\diamond, \mathcal{D}_n) \\ &= \sum_{j=1}^n \alpha_{n,j}^\diamond(\mathbf{x}; \Theta_1, \dots, \Theta_k, \mathbf{X}_1^\diamond, \dots, \mathbf{X}_n^\diamond, \\ & \quad W_1^\diamond, \dots, W_n^\diamond, \mathcal{D}_n) Y_j^{\diamond, new} \\ & \quad - \sum_{j=1}^n \alpha'_{n,j}(\mathbf{x}; \Theta_1, \dots, \Theta_k, \mathbf{X}_1^\diamond, \dots, \mathbf{X}_n^\diamond, \\ & \quad W_1^\diamond, \dots, W_n^\diamond, \mathcal{D}_n) Y_j^{\diamond, new}, \end{aligned}$$

where the weights are, for any $j = 1, \dots, n$,

$$\begin{aligned} & \alpha_{n,j}^\diamond(\mathbf{x}; \Theta_1, \dots, \Theta_k, \mathbf{X}_1^\diamond, \dots, \mathbf{X}_n^\diamond, W_1^\diamond, \dots, W_n^\diamond, \mathcal{D}_n) \\ &= \frac{1}{B} \sum_{\ell=1}^B \frac{\mathbb{1}\{\mathbf{x}_j^\diamond \in A_n(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)\} \cap W_j^\diamond = 1}{N_{n,1}^\diamond(\mathbf{x}; \Theta_\ell, \mathbf{X}_1^\diamond, \dots, \mathbf{X}_n^\diamond, W_1^\diamond, \dots, W_n^\diamond, \mathcal{D}_n)}. \end{aligned}$$

with $N_{n,1}^\diamond(\mathbf{x}; \Theta_\ell, \mathbf{X}_1^\diamond, \dots, \mathbf{X}_n^\diamond, W_1^\diamond, \dots, W_n^\diamond, \mathcal{D}_n)$, the number of elements of \mathcal{D}_n^\diamond that fall into $A_n(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)$ such that $W^\diamond = 1$. Throughout this section, we shall use the convention $\frac{0}{0} = 0$ in case $N_{n,1}^\diamond(\mathbf{x}; \Theta_\ell, \mathbf{X}_1^\diamond, \dots, \mathbf{X}_n^\diamond, W_1^\diamond, \dots, W_n^\diamond, \mathcal{D}_n) = 0$ and thus $\mathbb{1}\{\mathbf{x}_j^\diamond \in A_n(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)\} \cap W_j^\diamond = 1 = 0$ for $j = 1, \dots, n$. Similarly, we have:

$$\begin{aligned} & \alpha'_{n,j}(\mathbf{x}; \Theta_1, \dots, \Theta_k, \mathbf{X}_1^\diamond, \dots, \mathbf{X}_n^\diamond, W_1^\diamond, \dots, W_n^\diamond, \mathcal{D}_n) \\ &= \frac{1}{B} \sum_{\ell=1}^B \frac{\mathbb{1}\{\mathbf{x}_j^\diamond \in A_n(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)\} \cap W_j^\diamond = 0}{N_{n,0}^\diamond(\mathbf{x}; \Theta_\ell, \mathbf{X}_1^\diamond, \dots, \mathbf{X}_n^\diamond, W_1^\diamond, \dots, W_n^\diamond, \mathcal{D}_n)}. \end{aligned}$$

To lighten the notation in the sequel, we will simply write:

$$\begin{aligned} \tau_{B,n}^{new, \diamond}(\mathbf{x}) &= \sum_{j=1}^n \alpha_j^\diamond(\mathbf{x}) Y_j^{\diamond, new} - \sum_{j=1}^n \alpha'_j(\mathbf{x}) Y_j^{\diamond, new} \\ &= \tau_1^{new, \diamond}(\mathbf{x}) - \tau_0^{new, \diamond}(\mathbf{x}). \end{aligned}$$

In other words, the \mathcal{D}_n^\diamond sample allows the forest machine and the estimation process to be disconnected in the analysis of the algorithm.

Let $\mathbf{x} \in \mathcal{X}$, we have:

$$\begin{aligned} |\hat{\tau}^{new}(\mathbf{x}) - \tau^t(\mathbf{x})| &\leq |\hat{\tau}^{new}(\mathbf{x}) - \tau^{new,\diamond}(\mathbf{x})| \\ &\quad + |\tau^{new,\diamond}(\mathbf{x}) - \tau^t(\mathbf{x})|, \end{aligned}$$

and

$$\begin{aligned} |\tau^{new,\diamond}(\mathbf{x}) - \tau^t(\mathbf{x})| &\leq |\tau_1^{new,\diamond}(\mathbf{x}) - \tau_1^t(\mathbf{x})| \\ &\quad + |\tau_0^{new,\diamond}(\mathbf{x}) - \tau_0^t(\mathbf{x})|. \end{aligned}$$

Remark that

$$\tau_1^t(\mathbf{x}) = \mathbb{E}(Y^t(1)|\mathbf{X}^t = \mathbf{x}) = \mathbb{E}(Y^\diamond(1)|\mathbf{X}^\diamond = \mathbf{x}).$$

Each of the two terms will be treated the same way.

$$\begin{aligned} |\tau_1^{new,\diamond}(\mathbf{x}) - \tau_1^t(\mathbf{x})| &\leq \\ &\left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond(\mathbf{x}) [(Y_i^{new,\diamond}) - \mathbb{E}[Y^\diamond(1)|\mathbf{X}^\diamond = \mathbf{X}_i^\diamond]] \right| \\ &+ \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond(\mathbf{x}) [\mathbb{E}[Y^t(1)|\mathbf{X}^t = \mathbf{X}_i^\diamond] - \mathbb{E}[Y^t(1)|\mathbf{X}^t = \mathbf{x}]] \right| \\ &=: U_n + V_n. \end{aligned}$$

The convergence of U_n and V_n to 0 in L^1 is addressed in Appendix A. \square

5 Simulation results

In the following examples, causal offset with unique and distinct models are compared to the baseline case where HTERF is simply trained on the available target data. Two choices of algorithms are considered to estimate functions f^s and f^o , namely Kernel Ridge Regression and Regression Random Forest.

5.1 One-dimensional example

First, we consider a simple one-dimensional example where the source domain is defined by:

$$\begin{aligned} \mathbf{X}^s &\sim U([0, 1]), \quad W^s \sim \text{Bern}(0.5), \text{ and} \\ Y^s &= \sin(\mathbf{X}^s) \cdot W^s + \cos(\mathbf{X}^s). \end{aligned}$$

The target domain is defined by:

$\mathbf{X}^t \sim U([0, 1])$, $W^t \sim \text{Bern}(0.5)$, and
 $Y^t = \cos(\mathbf{X}^t) \cdot W^t + \cos(\mathbf{X}^t)$.

A first simulation is performed with a source sample of size 10,000 and target samples of size 500. The results are recorded in Table 1.

Method	RMSE
HTERF on source	0.003
Offset KRR separate models	0.015
Offset KRR unique model	0.009
No transfer (HTERF on target only)	0.205

Table 1: One-dimensional example. Root mean squared errors of CATE on source and on target with three different methods: offset causal with separate KRR models, offset causal with unique KRR model, and HTERF only trained on target data (baseline method). HTERF causal forests have 500 trees. The results are aggregated over 50 simulation replications with 500 test points each (the source dataset stays unchanged; only the target training and test datasets are modified).

Table 1 shows that both causal offset methods have better performance than the baseline method. In this example, using a single KRR model for treated and untreated individuals is more efficient than using two KRR models.

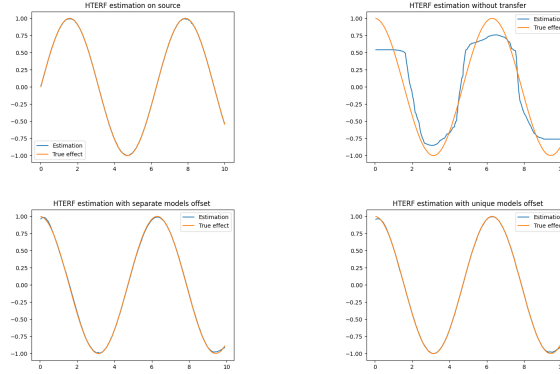


Figure 1: Graphical illustration for the one-dimensional example. First: HTERF CATE estimation on source. Second: HTERF CATE estimation on target using only target data. Third: HTERF CATE estimation on target using offset causal with separate KRR models. Fourth: HTERF CATE estimation on target using offset causal with unique KRR model.

In order to explore the sensitivity of the method to various parameters (leaf size, number of trees, sample size), we conducted a more extensive study. In order to help reproducibility of our work, the `Python` code used

to get these results is available at https://plmlab.math.cnrs.fr/maume/transfer_causal/. We keep the same source and target models as above. The offset is done with regression random forests. The graphs below (Figure 2) show the impact of leaf size, while Table 2 gives more insights on other parameters. Following Hill (2011); Hahn et al. (2020), we use PEHE (*precision in estimating heterogeneous effects*): mean squared error of CATE estimates for each unit in a dataset, as an error measure.

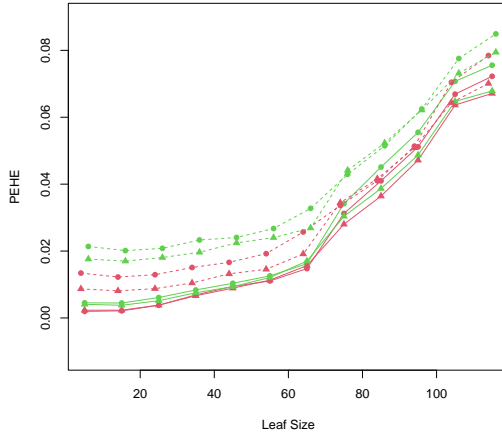


Figure 2: Simple example in dimension 1. PEHE are computed over 200 test points and the mean is taken over 30 repetitions; the source sample size is 5,000. The red lines are for two offset models; the green lines are for one single offset model. Solid lines correspond to a ratio of target sample size over source sample size of 5%, and the dotted lines correspond to a ratio of 2.5%. The lines with round markers are for 500 trees, while the lines with triangles are for 200 trees.

Figure 2 and Table 2 illustrate the convergence results with respect to the sample sizes, the influence of the leaf sizes, and the number of trees. Some cross-validation procedure could be used to select the best parameters, but it has not been implemented at this time. We remark on this one-dimensional example that the two offset models estimated by random forests give better results than the single random forest offset models.

5.2 Multi-dimensional example

Let us consider a multi-dimensional example inspired by the previous one. The source domain is:

$$\mathbf{X}^s = (\mathbf{X}^{s(1)}, \dots, \mathbf{X}^{s(10)}) \sim U([0, 1]^{10}),$$

Source (leaf)	Tree	Target/Source = 2,5%			Target/Source = 5%		
		Double	Single	Naive	Double	Single	Naive
10 000 (25)	100	0.249 (0.164)	0.397 (0.338)	51.391 (0.798)	0.062 (0.013)	0.073 (0.030)	40.460 (3.062)
	200	0.201 (0.075)	0.265 (0.115)	53.478 (0.818)	0.071 (0.019)	0.071 (0.031)	41.581 (4.32)
	300	0.234 (0.138)	0.394 (0.269)	53.101 (0.478)	0.068 (0.017)	0.113 (0.018)	43.253 (1.903)
	400	0.208 (0.115)	0.318 (0.189)	54.198 (0.366)	0.070 (0.017)	0.077 (0.018)	40.974 (3.450)
	500	0.209 (0.154)	0.332 (0.268)	52.942 (0.716)	0.061 (0.016)	0.067 (0.021)	40.543 (2.839)
5 000 (15)	100	1.359 (1.936)	2.427 (1.951)	55.855 (1.423)	0.178 (0.047)	0.288 (0.129)	51.248 (1.398)
	200	0.812 (0.499)	1.699 (0.908)	58.252 (2.654)	0.232 (0.209)	0.379 (0.272)	49.022 (2.017)
	300	1.323 (2.179)	2.562 (2.508)	53.016 (2.651)	0.203 (0.106)	0.339 (0.205)	49.299 (1.631)
	400	0.705 (0.547)	1.623 (0.827)	53.869 (1.389)	0.168 (0.066)	0.232 (0.099)	47.646 (1.048)
	500	1.224 (2.378)	2.013 (2.418)	52.781 (2.061)	0.213 (0.120)	0.447 (0.326)	48.280 (2.172)
2 000 (5)	100	6.02 (5.288)	16.094 (9.559)	56.484 (4.075)	1.477 (1.509)	2.516 (1.538)	41.795 (5.199)
	200	8.431 (6.310)	17.413 (10.330)	55.721 (7.077)	1.935 (2.383)	4.228 (4.213)	47.943 (2.709)
	300	8.377 (8.396)	17.535 (9.812)	55.632 (4.859)	1.863 (2.051)	3.664 (2.454)	48.548 (4.996)
	400	7.173 (5.630)	16.788 (8.099)	55.108 (7.145)	1.323 (1.396)	3.321 (3.118)	44.357 (3.714)
	500	6.567 (4.890)	18.326 (8.704)	55.767 (4.563)	2.336 (2.861)	3.598 (2.646)	49.795 (7.523)

Table 2: For the simple dimension 1 example, we have recorded the mean over 30 repetitions of PEHE computed over 200 test points; in brackets is the standard deviation. The computations have been done for different sample sizes (source and target), number of trees, and the leaf size is in brackets. The “Double” columns refer to the method with two offset models; the “Single” columns refer to the model with one offset model; the “Naive” columns refer to the HTERF algorithm on the target sample (i.e., with no transfer). Means and standard deviations are multiplied by 100.

$W^s \sim \text{Bern}(0.5)$, and
 $Y^s = \sin(\mathbf{X}^{s(1)})W^s + \cos(\mathbf{X}^{s(2)})$. The target domain is defined by: $\mathbf{X}^t \sim U([0, 1]^{10})$,
 $W^t \sim \text{Bern}(0.5)$, and
 $Y^t = \cos(\mathbf{X}^{t(1)})W^t + \cos(\mathbf{X}^{t(2)})$.

A first simulation is done with a source sample of size 10,000 and target samples of size 500.

Method	RMSE
Source	0.004
Offset separate KRR models	0.957
Offset unique KRR model	0.960
Offset separate RF models	0.120
Offset unique RF model	0.135
No transfer	0.348

Table 3: Multi-dimensional example. Root mean squared errors of CATE on source and on target with five different methods: offset causal with separate KRR models, offset causal with unique KRR model, offset causal with separate random forest (RF) models, offset causal with unique RF model, and HTERF only trained on target data (baseline method). HTERF causal forests have 500 trees. The results are aggregated over 50 simulation replications with 500 test points each (the source dataset stays unchanged; only the target training and test datasets are modified).

Figure 3 illustrates the poor performance of KRR in estimating f^s and f^o . In the top right image, the KRR estimator of the function f^s fails to capture the variations of Y_0^s against $\mathbf{X}^{s(2)}$. However, when using random forests to estimate f^s and f^o , causal offset algorithms are more efficient than the baseline method (see Table 3). In this setting, using separate models for treated and control units is the best strategy.

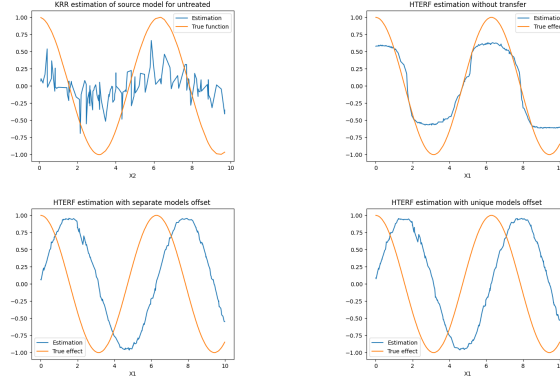


Figure 3: Graphical illustration for the multi-dimensional example. First: KRR estimation on source of the function f_0^s . Second: HTERF CATE estimation on target using only target data. Third: HTERF CATE estimation on target using offset causal with separate KRR models. Fourth: HTERF CATE estimation on target using offset causal with unique KRR model.

As in the one-dimensional case, we conducted a more extensive study to assess the impact of sample sizes, leaf sizes, and the number of trees on the results. The performance measure used is again PEHE. The offset is calculated using regression random forests. Figure 4 illustrates the impact of leaf size, while Table 4 provides further insights into the other parameters.

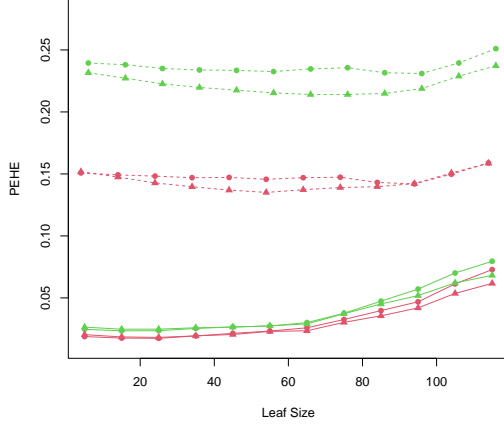


Figure 4: Example in higher dimension. PEHE are computed over 200 test points and the mean is taken over 30 repetitions; the source sample size is 5,000. The red lines are for two offset models; the green lines are for one single offset model. Solid lines correspond to a ratio of target sample size over source sample size of 5%, and the dotted lines correspond to a ratio of 2.5%. The lines with round markers are for 500 trees, while the lines with triangles are for 200 trees.

Remark that in this dimension 10 model, only the first two variables are used in the model. The eight remaining variables may be seen as noise. Thus, this example shows the robustness of our algorithms to nuisance variables. It should also be noted that, in this example, for source sample sizes and target sample sizes that are too small, applying the transfer method yields worse results than the naive method (without transfer), see Table 4.

5.3 A more complex model: Ishigami-like

We conclude this simulation study with a more complex and non-linear model: an Ishigami-like model in dimension 3. The source domain is:

$$\mathbf{X}^s = (\mathbf{X}^{s(1)}, \mathbf{X}^{s(2)}, \mathbf{X}^{s(3)}) \sim U([0, 1]^3), \\ W^s \sim \text{Bern}(0.5), \text{ and}$$

$$Y^s = \sin(\mathbf{X}^{s(1)}) + \left[\sin(\mathbf{X}^{s(2)})^2 \right] \cdot W^s \\ + 0.1[\mathbf{X}^{s(3)}]^2 \cdot \sin(\mathbf{X}^{s(1)}).$$

The target domain is defined by:

$$\mathbf{X}^t \sim U([0, 1]^3), W^t \sim \text{Bern}(0.5), \text{ and}$$

$$Y^t = \sin(\mathbf{X}^{t(1)}) + \left[\cos(\mathbf{X}^{t(2)})^2 \right] \cdot W^t \\ + 0.1[\mathbf{X}^{t(3)}]^2 \cdot \sin(\mathbf{X}^{t(1)}).$$

We conducted a study illustrating the impact of sample sizes, leaf sizes, and number of trees on the results. The performance measure is again PEHE. The offset is done with regression random forests. Figure 5 shows the impact of leaf size; Table 5 gives more insights on other parameters. For this

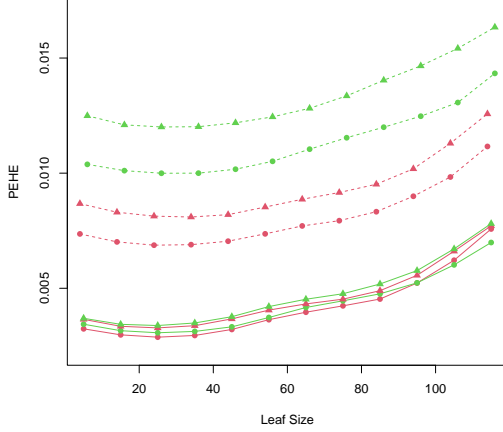


Figure 5: An Ishigami-like example. PEHE are computed over 200 test points and the mean is taken over 30 repetitions; the source sample size is 10,000. The red lines are for two offset models; the green lines are for one single offset model. Solid lines correspond to a ratio of target sample size over source sample size of 5%, and the dotted lines correspond to a ratio of 2.5%. The lines with round markers are for 300 trees, while the lines with triangles are for 100 trees.

Ishigami-like example, since it is more complex, we have done simulations for source sample sizes of 5,000 and 10,000; the target sizes are 2.5% or 5%. In this case, the results without transfer are very poor, which emphasizes the interest of the transfer method.

5.4 Semi-synthetic dataset

A final example is presented using the IHDP dataset, already introduced in the HTERF article Jocteur et al. (2024). This dataset has been studied in Wei et al. (2024) to compare the accuracy of various ATE estimators in a transfer learning context. In addition to RMSE, two additional indicators of performance for ATE estimation are added:

- ATE1: $\frac{1}{n_u} \left| \sum_{i=1}^{n_u} \hat{\tau}_{B,n}^{new}(\mathbf{X}_i^{tU}) - \tau^t(\mathbf{X}_i^{tU}) \right|$
- ATE2: $\frac{1}{n_u} \sum_{i=1}^{n_u} \left| \hat{\tau}_{B,n}^{new}(\mathbf{X}_i^{tU}) - \tau^t(\mathbf{X}_i^{tU}) \right|$

To create the source and target domains, the binary variable “The mother drank alcohol during pregnancy” has been used: the source domain consists

of all children whose mothers did not drink, and the target domain consists of the children whose mothers drank alcohol.

In this example, causal offset with a unique random forest model is the most efficient for CATE and ATE estimation, followed by causal offset with two separate models (see Table 6). Both algorithms outperform the baseline method without transfer.

6 Discussion

In this work, we have presented an algorithm to perform transfer learning on the causal inference problem. This approach combines the offset algorithm, already used on regression problems, and the HTERF causal forest. The combination of these two methods allows for a consistency result on the CATE estimation in the target domain. A generalization bound is also shown; these results rely on stronger assumptions than the classical HTERF consistency, especially regarding the number of trees in the forest.

Additional work could be done on the proof of consistency to lighten the assumptions. An almost sure convergence might also be obtained instead of an L^1 convergence. Also, from a practical point of view, a cross-validation procedure would be useful, especially to select the optimal leaf size, even if the results are quite good by taking any leaf size in a window from around 5 to 50. One issue in order to propose a cross-validation procedure is that the quantity of interest (CATE) is not observed, as it would be the case in regression. Some work would have to be done to find the appropriate criterion for cross-validation.

A Proof of results

Sequel of the proof of Theorem 2. The U_n term rewrites:

$$\begin{aligned} U_n &= \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond(\mathbf{x}) \left(\varepsilon_{1,i}^{s,\diamond} - E_1^o(\mathcal{D}, \tilde{\mathbf{X}}_i) \right) \right| \\ &\leq \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond \varepsilon_{1,i}^{s,\diamond} \right| + \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond E_1^o(\mathcal{D}, \tilde{\mathbf{X}}_i) \right|. \end{aligned}$$

Since Y^s, Y^t are assumed to be bounded, f^s, f^o are assumed to be continuous, and \mathbf{X} lives in a compact space, then ε^s and ε^t are bounded. Following the HTERF consistency proof (see the Appendix of Jocteur et al. (2024) and

the use of Lemma A.7 there), we have (where C denotes a generic positive constant):

$$\begin{aligned} \mathbb{E} \left[\left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond \varepsilon_{1,i}^{s,\diamond} \right|^2 \right] &\leq \|\varepsilon_1^s\|_\infty^2 \cdot \left(\frac{C}{\sqrt{n}(\ln n)^\beta} \right. \\ &\quad \left. + \frac{C^2}{n(\ln n)^\gamma} + C\sqrt{n}(n+1)^{2d}e^{-C(\ln n)^{2\beta}/C} \right) \\ &\rightarrow 0 \text{ as } n \rightarrow \infty. \end{aligned}$$

For the last term, we use the following decomposition:

$$\begin{aligned} \mathbb{E} \left[\alpha_i^\diamond \left| E_1^o(\mathcal{D}, \tilde{\mathbf{X}}_i) \right| \right] &\leq \mathbb{E} \left[\left| \alpha_i^\diamond - \frac{1}{n} \right| \left| E_1^o(\mathcal{D}, \tilde{\mathbf{X}}_i) \right| \right] \\ &\quad + \frac{1}{n} \mathbb{E} \left[\left| E_1^o(\mathcal{D}, \tilde{\mathbf{X}}_i) \right| \right] \end{aligned}$$

Since $\sum_{j=1}^n \alpha_j^\diamond = 1$ and the α_j^\diamond 's are identically distributed conditionally on \mathcal{D} , we have for any $n \in \mathbb{N}$: $\mathbb{E}[\alpha_i^\diamond | \mathcal{D}] = \frac{1}{n}$ and $\mathbb{E}[\alpha_i^\diamond] = \frac{1}{n}$. The Cauchy-Schwarz inequality gives:

$$\begin{aligned} \mathbb{E} \left[\left| \alpha_i^\diamond - \frac{1}{n} \right| \left| E_1^o(\mathcal{D}, \tilde{\mathbf{X}}_i) \right| \right] &\leq \sqrt{\text{Var}(\alpha_i^\diamond)} \sqrt{\mathbb{E} \left[\left| E_1^o(\mathcal{D}, \tilde{\mathbf{X}}_i) \right|^2 \right]}. \end{aligned}$$

Using the total variance formula:

$$\text{Var}(\alpha_i^\diamond) = \mathbb{E} [\text{Var}(\alpha_i^\diamond | \mathcal{D})].$$

We can rewrite:

$$\alpha_i^\diamond = \frac{1}{B} \sum_{l=1}^B \frac{\mathbb{1}_{\mathbf{X}_i^\diamond \in A_n(l)}}{N_{n,1}^\diamond(\mathbf{x}; \Theta_l, \mathcal{D}_n)} =: \frac{1}{B} \sum_{l=1}^B Z_l,$$

where conditionally on \mathcal{D} , the $(Z_l)_{l \in \{1, \dots, n\}}$ are independent and identically distributed; this leads to

$$\begin{aligned}
\text{Var}(\alpha_i^\diamond | \mathcal{D}) &= \frac{1}{B} \text{Var}(Z_1) \\
&\leq \frac{1}{B} \mathbb{E}[Z_1^2 | \mathcal{D}] \\
&\leq \frac{1}{B} \mathbb{E} \left[\frac{\mathbb{1}_{\mathbf{X}_i^\diamond \in A_n(1)}}{(N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n))^2} \middle| \mathcal{D} \right] \\
&\leq \frac{1}{B} \mathbb{E} \left[\frac{\mathbb{1}_{\mathbf{X}_i^\diamond \in A_n(1)} \mathbb{1}_{\{N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n) \geq \lambda\}}}{(N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n))^2} \middle| \mathcal{D} \right] \\
&\quad + \frac{1}{B} \mathbb{E} \left[\frac{\mathbb{1}_{\mathbf{X}_i^\diamond \in A_n(1)} \mathbb{1}_{\{N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n) < \lambda\}}}{(N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n))^2} \middle| \mathcal{D} \right].
\end{aligned}$$

Let $\lambda = \frac{\sqrt{n}(\ln n)^\beta}{2}$,

$$\begin{aligned}
\text{Var}(\alpha_1^\diamond | \mathcal{D}) &\leq \frac{1}{B\lambda} \mathbb{E} \left[\frac{\mathbb{1}_{\mathbf{X}_i^\diamond \in A_n(1)}}{N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n)} \middle| \mathcal{D} \right] \\
&\quad + \frac{1}{B} \mathbb{P} \left(N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n) < \lambda \middle| \mathcal{D} \right) \\
&\leq \frac{1}{B\lambda} \mathbb{E}[\alpha_i^\diamond | \mathcal{D}] \\
&\quad + \frac{1}{B} \mathbb{P} \left(N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n) < \lambda \middle| \mathcal{D} \right) \\
&\leq \frac{1}{B\lambda n} + \mathbb{P} \left(N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n) < \lambda \middle| \mathcal{D} \right).
\end{aligned}$$

Remark that

$$\begin{aligned}
&\left\{ N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n) < \frac{\sqrt{n}(\ln n)^\beta}{2} \right\} \\
&\subset \left\{ |N_{n,1}(\mathbf{x}; \Theta_1, \mathcal{D}_n) - N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n)| \right. \\
&\quad \left. > \frac{\sqrt{n}(\ln n)^\beta}{2} \right\},
\end{aligned}$$

thus we have

$$\begin{aligned}
&\mathbb{P} \left(N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n) < \lambda \middle| \mathcal{D} \right) \leq \\
&\mathbb{P} \left(|N_{n,1}(\mathbf{x}; \Theta_1, \mathcal{D}_n) - N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n)| > \lambda \middle| \mathcal{D} \right).
\end{aligned}$$

The following lemma has been stated in Jocteur et al. (2024), it is Lemma A.7 there; it is based on Vapnik-Chervonenkis theory.

Lemma 3. Consider $u \in \{0, 1\}$, as before, $N_{n,u}(A_n(\Theta)) = N_{n,u}(\mathbf{x}; \Theta, \mathcal{D}_n)$ is the number of observations of \mathcal{D}_n such that $W = u$ that fall into $A_n(\Theta) = A_n(\mathbf{x}; \Theta, \mathcal{D}_n)$, and $N_{n,u}^\diamond(A_n(\Theta)) = N_{n,u}^\diamond(\mathbf{x}; \Theta, \mathbf{X}^{\diamond 1}, \dots, \mathbf{X}^{\diamond n}, \mathcal{D}_n)$, the number of observations of \mathcal{D}_n^\diamond such that $W = u$ that fall into $A_n(\Theta)$. Then, $\forall \varepsilon > 0$,

$$\begin{aligned} \mathbb{P}(|N_{n,u}(A_n(\Theta)) - N_{n,u}^\diamond(A_n(\Theta))| > \varepsilon) \\ \leq 16(n+1)^{2d} e^{-\varepsilon^2/128n}. \end{aligned}$$

Using Assumption 2 and Lemma 3, there exist C and M positive constants such that:

$$\begin{aligned} \text{Var}(\alpha_i^\diamond) &\leq \frac{1}{B\lambda n} + \mathbb{E} \left[\mathbb{P}(|N_{n,1}(\mathbf{x}; \Theta_1, \mathcal{D}_n) - N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n)| > \frac{\sqrt{n}(\ln n)^\beta}{2} \middle| \mathcal{D}) \right] \\ &\leq \frac{2}{Bn^{3/2}(\ln n)^\beta} + \mathbb{P}(|N_{n,1}(\mathbf{x}; \Theta_1, \mathcal{D}_n) - N_{n,1}^\diamond(\mathbf{x}; \Theta_1, \mathcal{D}_n)| > \frac{\sqrt{n}(\ln n)^\beta}{2}) \\ &\leq \frac{2}{Mn^2} + 4C(n+1)^{2d} e^{-(\ln n)^{2\beta}/512} \\ &= \mathcal{O}\left(\frac{1}{n^2}\right). \end{aligned}$$

Now, since we assume that E_1^o converges to 0 in L^2 as n goes to ∞ , it also converges to 0 in L^1 :

$$\begin{aligned} \mathbb{E} \left[\left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond E_1^o(\mathcal{D}, \mathbf{X}_i^\diamond) \right| \right] &\leq \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \sqrt{\text{Var}(\alpha_i^\diamond)} \\ &\leq \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \sqrt{\mathbb{E}[|E_1^o(\mathcal{D}, \mathbf{X}_i^\diamond)|^2]} \\ &\quad + \mathbb{E}[|E_1^o(\mathcal{D}, \mathbf{X}_1^\diamond)|] \\ &\leq n \sqrt{\text{Var}(\alpha_1^\diamond)} \\ &\quad + \sqrt{\mathbb{E}[|E_1^o(\mathcal{D}, \mathbf{X}_1^\diamond)|^2]} \\ &\quad + \mathbb{E}[|E_1^o(\mathcal{D}, \mathbf{X}_1^\diamond)|] \\ &\longrightarrow 0 \text{ as } n \rightarrow \infty. \end{aligned}$$

The term V_n can now be treated:

$$V_n = \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond(\mathbf{x}) [\mathbb{E}(Y^t(1)|\mathbf{X}^t = \mathbf{X}_i^\diamond) - \mathbb{E}(Y^t(1)|\mathbf{X}^t = \mathbf{x})] \right|$$

The following lemma allows to control the variation of τ_1 on leaves. It is Lemma 2 in Meinshausen and Ridgeway (2006) and it is similar to Lemma 5 in Bénard et al. (2022).

Lemma 4. *Let Assumptions 1 and 2 be verified, let $\mathbf{x} \in \mathcal{X}$ and $\ell \in [1, B]$. Denote $A_n(\mathbf{x}, \Theta_\ell, \mathcal{D}_n) = \bigotimes_{j=1}^d I(\mathbf{x}, \Theta_\ell, \mathcal{D}_n)$, where $I(\mathbf{x}, \Theta_\ell, \mathcal{D}_n)$ are intervals; then,*

$$\max_{j=1, \dots, d} |I(\mathbf{x}, \Theta_\ell, \mathcal{D}_n)| = o(1).$$

Combining Lemma 4 with the continuity of τ_1 , we get

$$\forall \ell \in [1, B], \forall \mathbf{x} \in \mathcal{X}, \sup_{\mathbf{z} \in A_n(\mathbf{x}, \Theta_\ell, \mathcal{D}_n)} |\tau_1(\mathbf{z}) - \tau_1(\mathbf{x})| \xrightarrow[n \rightarrow \infty]{a.s.} 0.$$

Using this result, we get that V_n goes to 0 almost surely, as n goes to infinity: indeed, if for all $\ell = 1, \dots, B$ $\mathbf{X}_i^\diamond \notin A_n(\mathbf{x}, \Theta_\ell, \mathcal{D}_n)$ then $\alpha_i^\diamond(\mathbf{x}) = 0$ so that

$$\begin{aligned} & \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond(\mathbf{x}) [\mathbb{E}[Y^t(1)|\mathbf{X}^t = \mathbf{X}_i^\diamond] - \mathbb{E}[Y^t(1)|\mathbf{X}^t = \mathbf{x}]] \right| \\ & \leq \sup_{\mathbf{z} \in A_n(\mathbf{x})} |\tau_1(\mathbf{z}) - \tau_1(\mathbf{x})| \xrightarrow[n \rightarrow +\infty]{} 0. \end{aligned}$$

Since τ_1 is bounded, so is V_n and by the dominated convergence theorem, we have the L^1 convergence to 0 of V_n , when n goes to infinity.

It remains to consider the quantity $|\hat{\tau}^{new}(\mathbf{x}) - \tau^{new, \diamond}(\mathbf{x})|$. We consider separately but in a similar fashion $|\hat{\tau}_1^{new}(\mathbf{x}) - \tau_1^{new, \diamond}(\mathbf{x})|$ and $|\hat{\tau}_0^{new}(\mathbf{x}) - \tau_0^{new, \diamond}(\mathbf{x})|$:

$$\begin{aligned} \spadesuit &= |\hat{\tau}_1^{new}(\mathbf{x}) - \tau_1^{new, \diamond}(\mathbf{x})| = \\ & \left| \frac{1}{B} \sum_{l=1}^B \sum_{j=1}^n \frac{\mathbb{1}_{\mathbf{X}_j \in A_n(l)} \mathbb{1}_{W_j=1}}{N_{n,1}(\mathbf{x}; \Theta_l, \mathcal{D}_n)} Y_j^{new} - \right. \\ & \quad \left. \frac{\mathbb{1}_{\mathbf{X}_j^\diamond \in A_n(l)} \mathbb{1}_{W_j^\diamond=1}}{N_{n,1}^\diamond(\mathbf{x}; \Theta_l, \mathcal{D}_n)} Y_j^{new, \diamond} \right|. \end{aligned}$$

Following the HTERF consistency proof in Jocteur et al. (2024) (see the end of the Appendix there, where the similar quantity is considered), \spadesuit

converges to 0 almost surely, as n goes to infinity. The crucial point here is that the distributions $Y^{new}|\mathbf{X}^t, W^t$ and $Y^{new,\diamond}|\mathbf{X}^\diamond, W^\diamond$ are the same. As already mentioned, since all Y^t is bounded and f^s, f^o are continuous, we have that $Y^{new,\diamond}$ and Y^{new} are bounded. So that the dominated convergence theorem, implies that the \spadesuit term tends to 0 in L^1 as n goes to infinity. \square

B Generalisation bound

We start with the following decomposition:

$$\begin{aligned}
& \left| \hat{\tau}_1^{new}(\mathbf{x}) - \tau_1^t(\mathbf{x}) \right| \\
& \leq \left| \hat{\tau}_1^{new}(\mathbf{x}) - \tau_1^{new,\diamond}(\mathbf{x}) \right| + \underbrace{\left| \tau_1^{new,\diamond}(\mathbf{x}) - \tau_1^t(\mathbf{x}) \right|}_{\leq U_n + V_n} \\
& \leq \left| \frac{1}{B} \sum_{l=1}^B \sum_{j=1}^n \frac{\mathbb{1}_{\mathbf{X}_j \in A_n(l)} \mathbb{1}_{W_j=1}}{N_{n,1}(\mathbf{x}; \Theta_l, \mathcal{D}_n)} Y_j^{new} \right. \\
& \quad \left. - \frac{\mathbb{1}_{\mathbf{X}_j^\diamond \in A_n(l)} \mathbb{1}_{W_j^\diamond=1}}{N_{n,1}^\diamond(\mathbf{x}; \Theta_l, \mathcal{D}_n)} Y_j^{new,\diamond} \right| + \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond E_1^o(\mathcal{D}, \mathbf{X}_i^\diamond) \right| + \\
& \quad \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond \varepsilon_{1,i}^{s,\diamond} \right| + \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond(\mathbf{x}) \left[\mathbb{E}[Y^t(1)|\mathbf{X}^t = \mathbf{X}_i^\diamond] - \right. \right. \\
& \quad \left. \left. \mathbb{E}[Y^t(1)|\mathbf{X}^t = \mathbf{x}] \right] \right| \\
& \leq B_{offset} + B_{HTERF},
\end{aligned}$$

where

$$\begin{aligned}
B_{offset} = & \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond(\mathbf{x}) E_1^o(\mathcal{D}, \mathbf{X}_i^\diamond) \right| + \\
& \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond \varepsilon_{1,i}^{s,\diamond} \right|
\end{aligned}$$

and

$$B_{HTERF} = \left| \sum_{\substack{i=1 \\ W_i^\diamond=1}}^n \alpha_i^\diamond(\mathbf{x}) [\mathbb{E}[Y^t(1)|\mathbf{X} = \mathbf{X}_i^\diamond] - \mathbb{E}[Y^t(1)|\mathbf{X}^t = \mathbf{x}]] + \left| \frac{1}{B} \sum_{l=1}^B \sum_{j=1}^n \frac{\mathbb{1}_{\mathbf{X}_j \in A_n(l)} \mathbb{1}_{W_j=1}}{N_{n,1}(\mathbf{x}; \Theta_l, \mathcal{D}_n)} Y_j^{new} - \frac{\mathbb{1}_{\mathbf{X}_j^\diamond \in A_n(l)} \mathbb{1}_{W_j^\diamond=1}}{N_{n,1}^\diamond(\mathbf{x}; \Theta_l, \mathcal{D}_n)} Y_j^{new, \diamond} \right| \right|.$$

Acknowledgements

We are grateful to Natixis, specifically the Enterprise Risk Management department, for partially funding this work.

References

- Athey, S., Tibshirani, J., and Wager, S. (2019). Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178.
- Bénard, C., Da Veiga, S., and Scornet, E. (2022). Mean decrease accuracy for random forests: inconsistency, and a practical solution via the sobolmda. *Biometrika*, 109(4):881–900.
- Bica, I. and van der Schaar, M. (2022). Transfer learning on heterogeneous feature spaces for treatment effects estimation. *Advances in Neural Information Processing Systems*, 35:37184–37198.
- Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526.
- Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2006). A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19.
- Hahn, P. R., Murray, J. S., and Carvalho, C. M. (2020). Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion). *Bayesian Analysis*, 15(3):965–1056.
- Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240.

- Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. (2006). Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19.
- Huyen, C. (2022). *Designing machine learning systems*. O’Reilly Media, Inc., USA.
- Imbens, G. W. and Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, Cambridge, UK.
- Jocteur, B.-A. (2023). Julia package causalforest. jl. <https://hal.science/hal-04883933v1>.
- Jocteur, B.-A., Maume-Deschamps, V., and Ribereau, P. (2024). Heterogeneous treatment effect-based random forest: Hterf. *Computational Statistics & Data Analysis*, page 107970.
- Künzel, S. R., Stadie, B. C., Vemuri, N., Ramakrishnan, V., Sekhon, J. S., and Abbeel, P. (2018). Transfer learning for estimating causal effects using neural networks. *arXiv preprint arXiv:1808.07804*.
- Meinshausen, N. and Ridgeway, G. (2006). Quantile regression forests. *Journal of machine learning research*, 7(6).
- Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.
- Wahba, G. (2003). An introduction to reproducing kernel hilbert spaces and why they are so useful. In *Proceedings of the 13th IFAC Symposium on System Identification (SYSID 2003)*, pages 525–528.
- Wang, X. (2016). *Active Transfer Learning*. PhD thesis, Ph. D. Dissertation. BAE Systems.
- Wang, X., Huang, T.-K., and Schneider, J. (2014). Active transfer learning under model shift. In *International Conference on Machine Learning*, pages 1305–1313.
- Wang, X. and Schneider, J. G. (2015). Generalization bounds for transfer learning under model shift. In *UAI*, pages 922–931.
- Wei, S., Zhang, H., Moore, R., Kamaleswaran, R., and Xie, Y. (2024). Transfer learning for causal effect estimation.
- Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. (2013). Domain adaptation under target and conditional shift. In *International conference on machine learning*, pages 819–827.

Zhao, H., Des Combes, R. T., Zhang, K., and Gordon, G. (2019). On learning invariant representations for domain adaptation. In *International conference on machine learning*, pages 7523–7532.

Source	Tree (leaf)	Target/Source = 2,5%			Target/Source = 5%		
		Double	Single	Naive	Double	Single	Naive
10 000 (45)	100	1.637 (0.962)	2.061 (1.233)	56.753 (0.809)	0.461 (0.084)	0.514 (0.126)	55.237 (0.184)
	200	1.552 (0.462)	1.882 (0.723)	54.814 (0.305)	0.434 (0.088)	0.471 (0.097)	53.792 (0.683)
	300	1.722 (0.545)	2.240 (0.755)	51.135 (0.810)	0.428 (0.068)	0.476 (0.069)	51.218 (0.519)
	400	1.299 (0.406)	1.719 (0.487)	52.315 (0.700)	0.415 (0.087)	0.451 (0.107)	52.901 (0.643)
	500	1.635 (0.616)	2.251 (0.676)	53.689 (0.796)	0.046 (0.070)	0.496 (0.097)	52.982 (0.415)
31 5 000 (35)	100	17.379 (6.779)	27.096 (10.853)	53.626 (2.108)	1.866 (0.505)	2.568 (0.819)	50.090 (1.399)
	200	13.967 (10.830)	21.973 (11.071)	52.598 (1.627)	1.938 (0.889)	2.591 (0.933)	53.476 (0.662)
	300	13.713 (8.329)	20.674 (7.285)	59.665 (4.051)	2.074 (0.557)	3.013 (0.955)	53.985 (0.702)
	400	18.109 (12.446)	26.480 (11.622)	52.722 (1.908)	2.111 (0.671)	2.977 (0.597)	50.223 (0.979)
	500	14.697 (8.907)	23.383 (10.371)	51.288 (2.606)	1.934 (0.462)	2.535 (0.542)	53.296 (0.906)
2 000 (25)	100	66.089 (15.218)	77.807 (11.887)	56.293 (9.582)	24.284 (11.014)	39.270 (10.134)	51.797 (2.947)
	200	65.477 (12.587)	77.681 (7.363)	57.284 (9.536)	20.093 (9.438)	33.099 (9.842)	54.067 (3.345)
	300	56.928 (13.383)	68.575 (8.217)	51.632 (4.537)	26.992 (15.559)	38.181 (11.114)	53.646 (3.100)
	400	70.315 (17.692)	79.726 (8.729)	60.383 (7.795)	25.630 (15.875)	35.515 (11.915)	57.631 (3.967)
	500	53.397 (11.787)	67.583 (9.483)	54.560 (7.482)	26.673 (11.469)	36.297 (9.735)	51.801 (3.876)

Table 4: Example in higher dimension. We have recorded the mean over 30 repetitions of PEHE computed over 200 test points; in brackets is the standard deviation. The computations have been done for different sample sizes (source and target), number of trees, and the leaf size is in brackets. Means and standard deviations are multiplied by 100.

Source	Tree (leaf)	Target/Source = 2,5%			Target/Source = 5%		
		Double	Single	Naive	Double	Single	Naive
10 000 (35)	100	0.809 (0.333)	1.202 (0.619)	12.996 (0.184)	0.337 (0.069)	0.349 (0.080)	12.565 (0.141)
	300	0.689 (0.173)	1.000 (0.277)	11.915 (0.344)	0.295 (0.051)	0.313 (0.077)	11.356 (0.330)
	500	0.797 (0.244)	1.113 (0.437)	13.106 (0.254)	0.321 (0.060)	0.345 (0.059)	11.480 (0.225)
5 000 (25)	100	2.648 (1.975)	3.843 (2.226)	12.275 (0.236)	1.264 (0.281)	1.836 (0.689)	13.628 (0.417)
	300	2.745 (1.210)	4.256 (1.585)	13.006 (0.671)	1.219 (0.355)	1.823 (0.971)	12.062 (0.316)
	500	2.211 (0.764)	3.978 (1.369)	13.257 (0.222)	1.075 (0.392)	1.565 (0.691)	12.113 (0.163)

Table 5: An Ishigami-like model. We have recorded the mean over 30 repetitions of PEHE computed over 200 test points; in brackets is the standard deviation. The computations have been done for different sample sizes (source and target), number of trees, and the leaf size is in brackets. Means and standard deviations are multiplied by 100.

Method	CATE	ATE1	ATE2
Offset separate models	0.808	0.292	0.561
Offset unique model	0.734	0.180	0.491
No transfer	1.016	0.351	0.732

Table 6: RMSE on CATE and two different errors on ATE with three different methods: offset causal with separate RF models, offset causal with unique RF model, and HTERF only trained on target data (baseline method). HTERF causal forests have 500 trees; the forest of the first step in HTERF has 500 trees. The results are aggregated over 50 simulation replications; the source dataset stays unchanged, but for each replication, the target sample is modified.