Transfer learning for causal forest

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

Journal of Machine Learning Research 25 (2024) 1-21 Submitted 6/24; Revised ; Published 6/24

Transfer learning for causal forest

Bérénice-Alexia Jocteur BERENICE-ALEXIA.JOCTEUR@NATIXIS.COM
Universite Claude Bernard Lyon 1, CNRS, Ecole Centrale de Lyon, INSA Lyon, Université Jean Monnet, ICJ UMR5208, 69622 Villeurbanne, France
Enterprise Risk Management, Natixis, 75013 Paris France.

Véronique Maume-Deschamps VERONIQUE.MAUME@UNIV-LYON1.FR
Universite Claude Bernard Lyon 1, CNRS, Ecole Centrale de Lyon, INSA Lyon, Université Jean Monnet, ICJ UMR5208, 69622 Villeurbanne, France.

Pierre Ribereau PIERRE.RIBEREAU@UNIV-LYON1.FR
Universite Claude Bernard Lyon 1, CNRS, Ecole Centrale de Lyon, INSA Lyon, Université Jean Monnet, ICJ UMR5208, 69622 Villeurbanne, France.

Editor: My editor TODO

Abstract

Transfer learning addresses the challenge of transfering knowledge from one domain to another. Traditional transfer learning focuses on adapting models trained on a source domain (with a lot of observations) to improve performance on a target domain (with few observations). In this work we consider the case of a model shift and we focus on the 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 in order to estimate the offset between source and target distributions. Our main result is a bound on the CATE error of HTERF on target depending on the error of the intermediate models. Simulation studies show the good performances of this approach in different settings on simulations and on a real-world dataset.

Keywords: causal inference, causal forest, HTERF, transfer learning, domain adaptation, offset method

1 Introduction

Estimates of causal effects are needed to answer what-if questions about shifts in policies, such as new treatments in pharmacology or new pricing strategies for business owners. This section presents the main required notions: the potential outcome framework, the Heterogneous Treatment Effect based Random Forest - HTERF - method for Conditional Average Treatment Effect - CATE - estimation and transfer learning, with a focus on domain adaptation. The paper is organised as follows. In Section 2, we give details on the offset method. Section 3 contains our adaptation of the offset method to CATE estimation and our L^1 consistency result, a generalisation bound is also obtained. A simulations study is presented in Section 4. A short discussion can be found in Section 5. Most technical proofs are postponed to the final Appendix.

1.1 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 outcome 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:

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

The average treatment effect (ATE) is:

$$\tau = \mathbb{E}\left[Y(1) - Y(0)\right] \tag{2}$$

A standard assumption for 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 :

$$\{Y(1), Y(0)\} \perp\!\!\!\perp W | \mathbf{X}. \tag{3}$$

Many algorithms in the literature allow to evaluate CATE: causal forests, metalearners, causal neural networks as examples. In what follows we focus on HTERF, a short presentation of this algorithm is given in the next section.

1.2 HTERF

The presented transfer algorithms are based on HTERF, a special case of random forest introduced by 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 $\mathcal{D}_n = (W_i, \mathbf{X}_i, Y_i)_{i=1,\dots,n} \in \{0,1\} \times \mathbb{R}^d \times \mathbb{R}$, it provides an estimator $\hat{\tau}_{B,n}(\mathbf{x})$ of the quantity $\tau(\mathbf{x})$. Given assumptions on the distribution of $\mathcal{D}_n$ and the construction of the forest, an almost sure convergence result of $\hat{\tau}_{B,n}$ to τ is obtained, as well as an interpretability result. This algorithm has been implemented in `Julia`, in the package `CausalForest`[1]. 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 metalearners on most of the tested settings.

1.3 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

1. `https://github.com/BereniceAlexiaJocteur/CausalForest.jl`

scarce. This approach allows models to generalize better and achieve improved performances, 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 all have the same feature space (but different distributions), while transfer learning includes cases where the target domain's feature space is different from the source feature space or spaces. In what follows, the problem of supervised domain adaptation is considered, where both source and target dataset 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 modelling $P(Y|\mathbf{X})$. 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 treated 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 $P(Y|\mathbf{X})$ stays 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 the model shift or concept drift concerns the 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 strategy and the simplest is to train models on large and rich datasets hoping that points following both source and target distribution will be present in this large dataset. This method requires to have access to large external datasets susceptible 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 take into account a certain type of shift, for example the kernel mean matching (KMM) method (Huang et al. (2006), Gretton et al. (2006)) allows to deal with covariate shift. Zhang et al. (2013) proposes an approach to correct both covariate shift and label shift without using labels from target distribution (unsupervised domain adaptation problem), in a similar fashion 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, they use labeled source data. Finally a third kind of approach to deal with data distribution shift is to retrain the model with labeled target data, either the model is retrained from scratch with both source and target data or the existing model trained on source resumes its training on target data. This second option named fine tuning is easily applicable on neural networks by using technics 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, that is $P(Y^t(1)|\mathbf{X}^t) \neq P(Y^s(1)|\mathbf{X}^s)$ 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 it 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. A recent work has been done to estimate ATE in a supervised domain adaptation setup in Wei et al. (2024), the nuisance parameters (such as the propensity score) are estimated using ℓ^1 regularised transfer learning, and then plugged in 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, using NN weights estimated from the source domain and freezing some of its layers before backpropagating through the unfrozen ones when training on 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, since it allows transfer learning on CATE estimation without using a neural network.

2 The offset approach

In what follows, we will be concerned with the offset approach that we describe now.

2.1 Presentation

Let $\mathcal{X} \in \mathbb{R}^d$ and $\mathcal{Y} \in \mathbb{R}$ 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 data set of size n, we also consider $(\mathbf{Z}_i^{tL})_{i\in\{1,\dots,n_l\}} = ((\mathbf{X}_i^{tL}, y_i^{tL}))_{i\in\{1,\dots,n_l\}}$ the labeled target data set of size n_l. There is also an unlabeled target dataset on which we want to test the performance of transfer learning we denote it $(\mathbf{X}_i^{tU})_{i\in\{1,\dots,n_u\}}$ of size n_u.

Algorithm 1 Offset algorithm

Input: A source data set $\{\mathbf{X}_i^s, Y_i^s\}_{i\in\{1,\dots,n\}}$, a labeled target data set $\{\mathbf{X}_i^{tL}, Y_i^{tL}\}_{i\in\{1,\dots,n_l\}}$ and an unlabeled target data set $\{\mathbf{X}_i^{tU}\}_{i\in\{1,\dots,n_u\}}$.
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 $\{\mathbf{X}_i^t\}$.
$\{Y_i^{new}\} \leftarrow \{Y_i^s + \hat{f}^o(\mathbf{X}_i^s)\}$
Train a model M on $\{\mathbf{X}_i^s, Y_i^{new}\} \cup \{\mathbf{X}_i^{tL}, Y_i^{tL}\}$.
Output: $\{\hat{Y}_i^{tU}\} \leftarrow \{M(\mathbf{X}_i^{tU})\}$

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

2.2 Using Kernel Ridge Regression

A generalisation bound is proposed in Wang and Schneider (2015) when Kernel Ridge Regression (KRR) is used in the offset algorithm.

KRR and its associated notations are defined the following way.

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 sample for a regression task in a reproducing kernel Hilbert space (see Wahba (2003)) $\mathcal{H}$ with kernel K, scalar product k and associated norm $||.||_k$. Let ℓ be the l^2 loss function, then the KRR estimator is:*

$$\arg\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(h, \mathbf{Z}_i) + \lambda||h||_k^2 \tag{4}$$

Two errors are defined:

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

Theorem 2 (Wang and Schneider (2015)) *If KRR is used to estimate the three functions in the offset method, let R^t be the generalisation error on the target dataset of the final model M, R^s_{emps} the empirical error on the source model and $\bar{R}^o_{emp}$ is the empirical error of the estimator $\hat{f}^o$ against $\{\mathbf{X}^{tL}, \hat{Y}^o\}$, then*

$$R^t - 2\left(R^s_{emps} - \bar{R}^o_{emp}\right) = O\left(\frac{1}{\sqrt{\lambda_o n_l}}\right), \tag{5}$$

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 than KRR (or only a weaker version of stability is obtained) which makes difficult extensions of this result to the causal case presented in the following section.

3 Causal adaptation

We propose an offset method adapted to the causal framework.

3.1 Overview

Two causal adaptation to the offset method are proposed, in Algorithm 2 the treated and the control populations are processed separately in order 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.

Algorithm 2 Offset causal algorithm separate models

Input: A source data set $\{W^s_i, \mathbf{X}^s_i, Y^s_i\}_{i\in\{1,\dots,n\}}$, a labeled target data set $\{W^{tL}_i, \mathbf{X}^{tL}_i, Y^{tL}_i\}_{i\in\{1,\dots,n_l\}}$ and an unlabeled target data set $\{\mathbf{X}^{tU}_i\}_{i\in\{1,\dots,n_u\}}$.

Estimate a model $\hat{f}^s_0$ that regresses $\{Y^s_i\}_{W^s_i=0}$ against $\{\mathbf{X}^s_i\}_{W^s_i=0}$ and a model $\hat{f}^s_1$ that regresses $\{Y^s_i\}_{W^s_i=1}$ against $\{\mathbf{X}^s_i\}_{W^s_i=1}$.

Estimate a model $\hat{f}^o_0$ that regresses $\{Y^{tL}_i - \hat{f}^s_0(\mathbf{X}^{tL}_i)\}_{W^{tL}_i=0}$ against $\{\mathbf{X}^{tL}_i\}_{W^{tL}_i=0}$ and a model $\hat{f}^o_1$ that regresses $\{Y^{tL}_i - \hat{f}^s_1(\mathbf{X}^{tL}_i)\}_{W^{tL}_i=1}$ against $\{\mathbf{X}^{tL}_i\}_{W^{tL}_i=1}$.

$\{Y^{new}_i\} \leftarrow \{Y^s_i + \hat{f}^o_{W^s_i}(\mathbf{X}^s_i)\}$

Train an HTERF model M on $\{W^s_i, \mathbf{X}^s_i, Y^{new}_i\} \cup \{W^{tL}_i, \mathbf{X}^{tL}_i, Y^{tL}_i\}$.

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

Any regression algorithm could be used to estimate the source and offset functions, in practice we obtained good results by using regression random forests.

Algorithm 3 Offset causal algorithm unique models

Input: A source data set $\{W_i^s, \mathbf{X}_i^s, Y_i^s\}_{i\in\{1,\dots,n\}}$, a labeled target data set $\{W_i^{tL}, \mathbf{X}_i^{tL}, Y_i^{tL}\}_{i\in\{1,\dots,n_l\}}$ and an unlabeled target data set $\{\mathbf{X}_i^{tU}\}_{i\in\{1,\dots,n_u\}}$.
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 $\{W_i^s, \mathbf{X}_i^s, Y_i^{new}\} \cup \{W_i^{tL}, \mathbf{X}_i^{tL}, Y_i^{tL}\}$.
Output: $\{\hat{\tau}^t(\mathbf{X}_i^{tU})\} \leftarrow \{M(\mathbf{X}_i^{tU})\}$

3.2 Convergence result

Since the size of the source dataset is assumed to be large compared to the target data set, we write a convergence theorem and a generalisation bound on the causal offset algorithm if the HTERF model in the last step is only fit on the set $\{W^s, \mathbf{X}^s, Y^{new}\} = \mathcal{D}_n$.

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 $\Theta = \left(\Theta^1, \Theta^2, \Theta^3\right)$ and independent of $\mathcal{D}_n$, and $\left(\Theta^1, \Theta^2\right)$ is independent of Θ^3. Θ^1 contains indices of observations that are used to build each tree. That is, the subsample $\mathcal{I}_1$, Θ^2 contains indices of observations that are used for estimations in each tree; namely, the subsample $\mathcal{I}_2$ and Θ^3 contains indices of splitting candidate variables in each node. We assume that Θ^3 gives a positive probability to each covariate. We must consider both Θ^1 and Θ^2 because $\mathcal{I}_2$ is the complementary of $\mathcal{I}_1$ in $\mathcal{I}$ (the subsample drawn before constructing a given tree) which is random itself.

- $\mathcal{D}_{n,1}^\star(\Theta_\ell)$ and $\mathcal{D}_{n,2}^\star(\Theta_\ell)$ 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 $\mathcal{D}_{n,2}^\star(\Theta_\ell)$ that fall into $A_n(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)$, such that $W_i = 1$ (resp. $W_i = 0$).

Let us consider $\tau_1^t(\mathbf{x}) = \mathbb{E}\left[Y^t(1)|\mathbf{X}^t = \mathbf{x}\right]$, $\tau_0^t(\mathbf{x}) = \mathbb{E}\left[Y^t(0)|\mathbf{X}^t = \mathbf{x}\right]$, $\hat{\tau}_1^{new}(\mathbf{x}) = \sum_{i:W_i=1} \alpha_i(\mathbf{x}) Y_i^{new}$ and $\hat{\tau}_0^{new}(\mathbf{x}) = \sum_{i:W_i=0} \alpha_i'(\mathbf{x}) Y_i^{new}$, where

$$\alpha_i(\mathbf{x}) = \frac{1}{B}\sum_{l=1}^{B} \frac{\mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x};\Theta_l,\mathcal{D}_n) \wedge W_i = 1 \wedge i \in \mathcal{D}_{n,2}^\star(\Theta_\ell)}}{N_{n,1}(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)}, \tag{6}$$

$$\alpha_i'(\mathbf{x}) = \frac{1}{B}\sum_{l=1}^{B} \frac{\mathbb{1}_{\mathbf{X}_i \in A_n(\mathbf{x};\Theta_l,\mathcal{D}_n) \wedge W_i = 0 \wedge i \in \mathcal{D}_{n,2}^\star(\Theta_\ell)}}{N_{n,0}(\mathbf{x}; \Theta_\ell, \mathcal{D}_n)}. \tag{7}$$

We shall make the following assumptions.

Assumption 1

- $Y^t = \tau^t(\mathbf{X}^t)g(\mathbf{W}^t) + \gamma^t(\mathbf{X}^t) + \varepsilon^t$ and $Y^s = \tau^s(\mathbf{X}^s)g(\mathbf{W}^s) + \gamma^s(\mathbf{X}^s) + \varepsilon^s$.
- $\forall i$ *such as* $W_i = 1, Y_i^s = f_1^s(\mathbf{X}_i^s) + \varepsilon_{1,i}^s, \varepsilon_1^s \perp\!\!\!\perp \mathbf{X}^s$ *and* $Y_i^t - f_1^s(\mathbf{X}_i^t) = f_1^o(\mathbf{X}_i^t) + \varepsilon_{1,i}^t, \varepsilon_1^t \perp\!\!\!\perp \mathbf{X}^s, \mathbf{X}^t$. ε_1^s *and* ε_1^t *are continuous centered random variables.*
- $\forall i$ *such as* $W_i = 0, Y_i^s = f_0^s(\mathbf{X}_i^s) + \varepsilon_{0,i}^s, \varepsilon_0^s \perp\!\!\!\perp \mathbf{X}^s$ *and* $Y_i^t - f_0^s(\mathbf{X}_i^t) = f_0^o(\mathbf{X}_i^t) + \varepsilon_{0,i}^t, \varepsilon_0^t \perp\!\!\!\perp \mathbf{X}^s, \mathbf{X}^t$. ε_0^s *and* ε_0^t *are continuous centered random variables.*
- $\mathbf{X}^s$ *and* $\mathbf{X}^t$ *are distributed as* $\mathbf{X} = (X_1, \ldots, X_d)$*, 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.*
- W^s *and* W^t *are distributed as* W*, a binary variable.*
- $\mathbf{X}$ *takes its values in* $\mathcal{X}$ *which is assumed to be a positive compact hyper-rectangle of* $\mathbb{R}^d$*:* $\mathcal{X} = \prod_{i=1}^{d} [u_i, v_i]$, $0 \leq u_i \leq v_i < \infty$.
- $\mathbf{x} \mapsto \gamma^t(\mathbf{x})$, $\mathbf{x} \mapsto \tau_1^s(\mathbf{x})$, $\mathbf{x} \mapsto \tau_0^s(\mathbf{x})$, $\mathbf{x} \mapsto \tau_1^t(\mathbf{x})$ *and* $\mathbf{x} \mapsto \tau_0^t(\mathbf{x})$ *are continuous. So in particular* $\mathbf{x} \mapsto \tau^t(\mathbf{x})$ *and* $\mathbf{x} \mapsto \tau^s(\mathbf{x})$ *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. $B = \mathcal{O}(\sqrt{n})$ and $\exists C > 0$ such as $B > C \dfrac{\sqrt{n}}{(\ln(n))^{\beta}}$, with $\beta > 1$.
2. $\forall \mathbf{x} \in \mathcal{X}, \quad \mathbb{E}[N_{n,1}(\mathbf{x}; \Theta, \mathcal{D}_n)] = \Omega\left(\sqrt{n}(\ln(n))^{\beta}\right)$.
3. $\forall \mathbf{x} \in \mathcal{X}, \quad \mathbb{E}[N_{n,0}(\mathbf{x}; \Theta, \mathcal{D}_n)] = \Omega\left(\sqrt{n}(\ln(n))^{\beta}\right)$.
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* $j = 1, \ldots, d$.
7. *HTERF as described in Jocteur et al. (2024) uses an honest framework, the training sample is splitted in two parts,* $\mathcal{I}_1$ *used to construct the splits of the trees and* $\mathcal{I}_2$ *is used to calculate the weights* α *and* α'*.* $\mathcal{I}_2$ *verifies that each split leaves at least a fraction* α *of the available training sample such as* $W = 1$ *(resp.* $W = 0$*) on each side of the split, for some* $0 < \alpha \leq 0.5$.

The function $\hat{f}_1^s$ is an estimator of f_1^s in the first step:

$$\forall i \text{ such as } W_i = 1, Y_i^s = \hat{f}_1^s(\mathbf{X}_i^s) + \epsilon_{1,i}^s + E_{1,i}^s(\mathcal{D}_s, \mathbf{X}_i^s) \tag{8}$$

The function $\hat{f}_1^o$ is an estimator of f_1^o in the second step:

$$\forall i \text{ such as } W_i = 1, Y_i^t - \hat{f}_1^s(\mathbf{X}_i^t) - E_1^s(\mathcal{D}_s, \mathbf{X}_i^t) = f_1^o(\mathbf{X}_i^t) + \epsilon_{1,i}^t \tag{9}$$

So:

$$\forall i \text{ such as } W_i = 1, Y_i^t - \hat{f}_1^s(\mathbf{X}_i^t) - E_1^s(\mathcal{D}_s, \mathbf{X}_i^t) = \hat{f}_1^o(\mathbf{X}_i^t) + E_1^o(\mathcal{D}, \mathbf{X}_i^t) + \epsilon_{1,i}^t \tag{10}$$

Finally the third step leads to:

$$\begin{aligned}
\forall i \text{ such as } W_i = 1, Y_i^{new} &= Y_i^s + \hat{f}_1^o(\mathbf{X}_i^s) \\
&= Y_i^s + Y_i^t - \hat{f}_1^s(\mathbf{X}_i^s) - E_1^s(\mathcal{D}_s, \mathbf{X}_i^s) - E_1^o(\mathcal{D}, \mathbf{X}_i^t) - \epsilon_{1,i}^t \\
&= Y_i^t + (Y_i^s - \hat{f}_1^s(\mathbf{X}_i^s)) - E_1^s(\mathcal{D}_s, \mathbf{X}_i^s) - E_1^o(\mathcal{D}, \mathbf{X}_i^t) - \epsilon_{1,i}^t \\
&= Y_i^t + \epsilon_{1,i}^s + E_1^s(\mathcal{D}_s, \mathbf{X}_i^s) - E_1^s(\mathcal{D}_s, \mathbf{X}_i^s) - E_1^o(\mathcal{D}, \mathbf{X}_i^t) - \epsilon_{1,i}^t \\
Y_i^{new} &= Y_i^t + \epsilon_{1,i}^s - \epsilon_{1,i}^t - E_1^o(\mathcal{D}, \mathbf{X}_i^t)
\end{aligned}$$

In a similar fashion we have:

$$\forall i \text{ such as } W_i = 0, Y_i^{new} = Y_i^t + \epsilon_{0,i}^s - \epsilon_{0,i}^t - E_0^o(\mathcal{D}, \mathbf{X}_i^t) \tag{11}$$

Then HTERF is trained on $\{W_i^s, \mathbf{X}_i^s, Y_i^{new}\}$, which gives the following estimator $\hat{\tau}_{B,n}^{new}(\mathbf{X}) = \hat{\tau}_1^{new}(\mathbf{x}) - \hat{\tau}_0^{new}(\mathbf{x})$, where $\hat{\tau}_1^{new}(\mathbf{x}) = \sum_{i:W_i=1} \alpha_i(\mathbf{x}) Y_i^{new}$ and $\hat{\tau}_0^{new}(\mathbf{x}) = \sum_{i:W_i=0} \alpha_i'(\mathbf{x}) Y_i^{new}$.

Theorem 3 *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 $\mathbb{E}\left[\max\left(\varepsilon_{1,i}^s\right)^2\right], \mathbb{E}\left[\max\left(\varepsilon_{1,i}^t\right)^2\right], \mathbb{E}\left[\max\left(\varepsilon_{0,i}^s\right)^2\right], \mathbb{E}\left[\max\left(\varepsilon_{0,i}^t\right)^2\right] \leq K(\ln n)^u$ with $\beta - u > \frac{1}{2}$ and K is a positive constant. Also assume that Y and E^o error term are bounded and that E^o converges to 0 in L^2 as n_l tends to ∞. Then*

$$\mathbb{E}\left[\left|\hat{\tau}_{B,n}^{new}(\mathbf{X}) - \tau^t(\mathbf{X})\right|\right] \underset{n, n_l \to \infty}{\longrightarrow} 0.$$

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

Remark 4 *With an estimator $\hat{f}^s$ of the form $\sum \omega_i Y_i^s$, since Y is assumed to be bounded, so are $\hat{f}^s$ and E^s. With $\hat{f}^o$ of the form $\sum \omega_i (Y_i^t - \hat{f}^s(\mathbf{X}_i^t))$, the error term E^o is also bounded. Most of the classical regression algorithm provide estimators of this form: random forest, linear regression, neural network...*

Remark 5 *Following what is done in the proof of Theorem 3, the error on the estimation of τ_1 can be bounded this way (same rationale applies for τ_0), let $\mathbf{x} \in \mathcal{X}$:*

$$\left|\hat{\tau}_1^{new}(\mathbf{x}) - \tau_1^t(\mathbf{x})\right| \leq Bound_{offset} + Bound_{HTERF}. \tag{12}$$

Overall this bound tends to 0 as $n, n_l \to +\infty$, 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 this bound are présented in Appendix B.

Proof [Proof of Theorem 3]

This proof is partially inspired from the proof of Theorem 4.1 in Jocteur et al. (2024). Let us define a diamond dataset $\mathcal{D}^\diamond = (Y_i^\diamond, \mathbf{X}_i^\diamond, W_i^\diamond)_{i=1,\dots,n}$ that is a sample of $(Y^t, \mathbf{X}, W)$, being independent of $\mathcal{D}^t$ and $\mathcal{D}^s$. This new sample is used to build $(Y_i^{new,\diamond})_{i=1,\dots,n}$ using the estimators $\hat{f}^s$ and $\hat{f}^o$ previously build:

$$\forall i \text{ such as } W_i^\diamond = 1, Y_i^{s,\diamond} = \hat{f}_1^s(\mathbf{X}_i^\diamond) + \epsilon_{1,i}^{s,\diamond} + E_1^{s,\diamond}(\mathcal{D}_s, \mathbf{X}_i^\diamond) \tag{13}$$

$$Y_i^\diamond - \hat{f}_1^s(\mathbf{X}_i^\diamond) - E_1^{s,\diamond}(\mathcal{D}_s, \mathbf{X}_i^\diamond) = \hat{f}_1^o(\mathbf{X}_i^\diamond) + E_1^{o,\diamond}(\mathcal{D}, \mathbf{X}_i^\diamond) + \epsilon_{1,i}^{t,\diamond} \tag{14}$$

$$Y_i^{new,\diamond} = Y_i^\diamond + \epsilon_{1,i}^{s,\diamond} - \epsilon_1^{t,\diamond} - E_1^{o,\diamond}(\mathcal{D}, \mathbf{X}_i^\diamond) \tag{15}$$

As in the HTERF consistency proof, the trees are grown using $\mathcal{D}_n = \mathcal{D}^s \cup \mathcal{D}^t$, but the sample $\mathcal{D}_n^\diamond$ (independent of $\mathcal{D}_n$ and Θ) is used to define a dummy estimator

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

where the weights are

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

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 as $W^\diamond = 1$. 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:

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

To lighten the notation in the sequel, we will simply write $\tau_{B,n}^{new,\diamond}(\mathbf{x}) = \sum_{j=1}^n \alpha_j^\diamond(\mathbf{x}) Y_j^{\diamond,new} - \sum_{j=1}^n \alpha_j^{'\diamond}(\mathbf{x}) Y_j^{\diamond,new} = \tau_1^{new,\diamond}(\mathbf{x}) - \tau_0^{new,\diamond}(\mathbf{x})$.
Let $\mathbf{x} \in \mathcal{X}$, we have:

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

Let $\mathbf{x}$ in $\mathcal{X}$: $\left|\tau^{new,\diamond}(\mathbf{x}) - \tau^t(\mathbf{x})\right| \leq \left|\tau_1^{new,\diamond}(\mathbf{x}) - \tau_1^t(\mathbf{x})\right| + \left|\tau_0^{new,\diamond}(\mathbf{x}) - \tau_0^t(\mathbf{x})\right|$. Each of the two terms will be treated the same way.

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

The convergence of U_n and V_n towards 0 are treated in Appendix A. ■

4 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.

4.1 One dimensional example

Firstly, the source domain is defined, let $\mathbf{X}_i^s \sim U([0,1])$, $W_i^s \sim Bern(0.5)$ and $Y_i^s = \tau^s(\mathbf{X}_i^s)W_i^s + \gamma^s(\mathbf{X}_i^s)$. Where $\tau^s(\mathbf{x}) = \sin(\mathbf{x})$ and $\gamma^s(\mathbf{x}) = \cos(\mathbf{x})$. A source sample of 10000 units in considered. The target domain is defined as $\mathbf{X}_i^t \sim U([0,1])$, $W_i^t \sim Bern(0.5)$ and $Y_i^t = \tau^t(\mathbf{X}_i)W_i + \gamma^t(\mathbf{X}_i)$. Where $\tau^t(\mathbf{x}) = \cos(\mathbf{x})$ and $\gamma^t(\mathbf{x}) = \cos(\mathbf{x})$. The unlabeled target dataset and the labeled target dataset are both of size 500.
In Table 1, the performance of HTERF on source model is presented in the first line. Then we present the results of the offset method with KRR used to estimate the functions f^s and f^o. Two cases have been considered in the first one, separate models are trained respectively for treated and control groups, in the second case the treatment variable T is considered as a covariate for f^s and f^o. Finally a no transfer strategy is considered where HTERF is trained only on the target data set. Figure 1 offers a graphical illustration of this example.

Both causal offset methods have better performances than the baseline method. In this example using a single model for treated and untreated individuals is the most efficient.

4.2 Multi-dimensional example

A multi-dimensional example inspired by the previous one is proposed, for the source domain let $\mathbf{X}_i^s \sim U([0,1]^{10})$, $W_i^s \sim Bern(0.5)$ and $Y_i^s = \tau^s(\mathbf{X}_i^s)W_i^s + \gamma^s(\mathbf{X}_i^s)$. Where $\tau^s(\mathbf{x}) = \sin(\mathbf{x}^{(1)})$ and $\gamma^s(\mathbf{x}) = \cos(\mathbf{x}^{(1)})$. A source sample of 10000 units in considered. The target domain is defined as $\mathbf{X}_i^t \sim U([0,1]^{10})$, $W_i^t \sim Bern(0.5)$ and $Y_i^t = \tau^t(\mathbf{X}_i)W_i + \gamma^t(\mathbf{X}_i)$. Where $\tau^t(\mathbf{x}) = \cos(\mathbf{x}^{(1)})$ and $\gamma^t(\mathbf{x}) = \cos(\mathbf{x}^{(2)})$. The unlabeled target dataset and the la-

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

Table 1: Dimension one example. Root mean squared errors of CATE on source and on target with three different methods namely, 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 forest of the first step in HTERF have 500 trees. The results are aggregated over 50 simulation replications with 500 test points each (the source dataset stay unchanged only the target training and test dataset are modified).

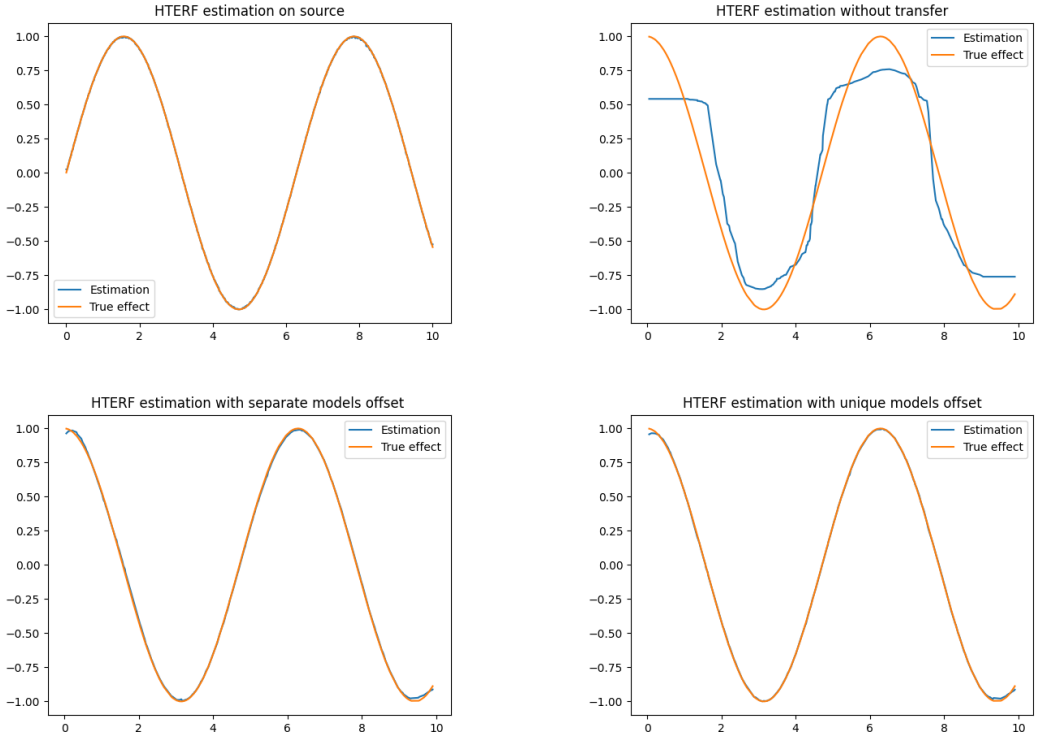

Figure 1: Graphical illustration for one dimensional example
Top left: HTERF CATE estimation on source
Top right: HTERF CATE estimation on target using only target data
Bottom left: HTERF CATE estimation on target using offset causal with separate KRR models
Bottom left: HTERF CATE estimation on target using offset causal with unique KRR model

beled target dataset are both 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 2: Multi-dimensional example. Root mean squared errors of CATE on source and on target with five different methods namely, 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 forest of the first step in HTERF have 500 trees. The results are aggregated over 50 simulation replications with 500 test points each (the source dataset stay unchanged only the target training and test datasets are modified).

Figure 2 illustrate the poor performance of KKR as the algorithm for 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 using random forest to estimate f^s and f^o, causal offset algorithms are more efficient than the baseline method (see Table 2). This time using separate models for treated and control units is the best strategy.

4.3 Semi-synthetic dataset

A last 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 accuracy of various ATE estimators in a transfer learning context. In addition to RMSE two additional indicators of performances for ATE estimation are added, namely:

- 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 whom mothers did not drink and the target domain consists of the children whom 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 3). Both algorithms outperform the baseline method without transfer.

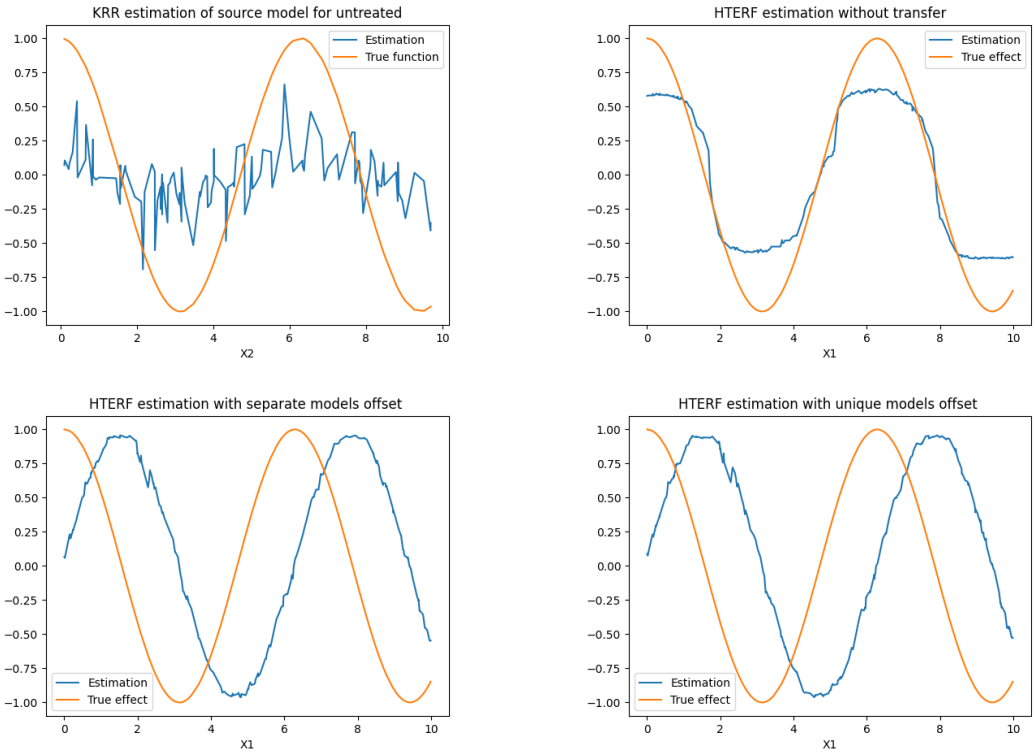

Figure 2: Graphical illustration for multi-dimensional example
Top left: KRR estimation on source of the function f_0^s
Top right: HTERF CATE estimation on target using only target data
Bottom left: HTERF CATE estimation on target using offset causal with separate KRR models
Bottom left: HTERF CATE estimation on target using offset causal with unique KRR model

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 3: RMSE on CATE and two different errors on ATE with three different methods namely, 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 have 500 trees. The results are aggregated over 50 simulation replications, the source dataset stays unchanged but for each replication the labeled target and unlabeled target are modified.

5 Discussion

We have presented in this work 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 to have a consistency result on the CATE estimation in the target domain. A generalisation bound is also shown, these results rely on stronger assumptions than the classical HTERF consistency, especially regarding the number of trees in the forest which needs to be large ($> C\frac{\sqrt{n}}{(\ln n)^\beta}$).
Additional work could be done on the proof on consistency to lighten the assumptions. An almost sure convergence might also be obtained instead of a L^1 convergence.

Appendix A. Proof of results

Proof [Sequel of the proof of Theorem 3]

The U_n term writes:

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

Since Y is assumed to be bounded, in addition to the fact that τ_1 and γ are continuous and $\mathbf{X}$ lives in a compact space, necessarily ϵ_1^s and ϵ_1^t are bounded. Following the HTERF consistency proof we have:

$$\begin{aligned} \mathbb{E}\left[\left| \sum_{\substack{i=1 \\ W_i^\diamond = 1}}^{n} \alpha_i^\diamond \varepsilon_{1,i}^{\diamond,s} \right|^2 \right] &\leq \mathbb{E}\left[\max \varepsilon_j^{\diamond 2} \right] \left(\frac{4}{K\sqrt{n}(\ln n)^\beta} + \frac{4CM^2}{n(\ln n)^\gamma} + 16C\sqrt{n}(n+1)^{2d} e^{-K^2(\ln n)^{2\beta}/2048} \right) \\ &\leq \frac{4C}{K\sqrt{n}(\ln n)^\beta} + \frac{4CM^2}{n(\ln n)^\gamma} + 16C\sqrt{n}(n+1)^{2d} e^{-K^2(\ln n)^{2\beta}/2048} \\ &\to 0 \end{aligned}$$

For the last term since the assumption is made that $E_1^o(\mathcal{D}, \mathbf{X}^s)$ is L^2 consistent, then it is also L^1 consistent, it can be bounded the following way

$$\mathbb{E}\left[\left| \sum_{\substack{i=1 \\ W_i^\diamond = 1}}^{n} \alpha_i^\diamond E_1^{o,\diamond}(\mathcal{D}, \mathbf{X}_i^\diamond) \right| \right] \leq \mathbb{E}\left[\alpha_1^\diamond \left| E_1^{o,\diamond}(\mathcal{D}, \mathbf{X}_1^\diamond) \right| \right]. \tag{16}$$

We use the following decomposition:

$$\mathbb{E}\left[\alpha_1^\diamond \left| E_1^{o,\diamond}(\mathcal{D}, \mathbf{X}_1^\diamond) \right| \right] \leq \mathbb{E}\left[\left| \alpha_i^\diamond - \frac{1}{n} \right| \left| E_1^{o,\diamond}(\mathcal{D}, \mathbf{X}_i^\diamond) \right| \right] + \frac{1}{n} \mathbb{E}\left[\left| E_1^{o,\diamond}(\mathcal{D}, \mathbf{X}_i^\diamond) \right| \right] \tag{17}$$

Since $\sum_{j=1}^n \alpha_j^\diamond = 1$ and the $\alpha_j^\diamond$s are identically distributed, we have $\mathbb{E}[\alpha_i^\diamond] = \frac{1}{n}$ and $\mathbb{E}[\alpha_i^\diamond | \mathcal{D}] = \frac{1}{n}$. Cauchy-Schwartz inequality gives:

$$\mathbb{E}\left[\left| \alpha_i^\diamond - \frac{1}{n} \right| \left| E_1^{o,\diamond}(\mathcal{D}, \mathbf{X}_i^\diamond) \right| \right] \leq \sqrt{Var(\alpha_i^\diamond)} \sqrt{\mathbb{E}\left[\left| E_1^{o,\diamond}(\mathcal{D}, \mathbf{X}_i^\diamond) \right|^2 \right]}. \tag{18}$$

Using the total variance formula:

$$Var(\alpha_i^\diamond) = \mathbb{E}\left[Var(\alpha_i^\diamond | \mathcal{D}) \right]. \tag{19}$$

We can rewrite:

$$\alpha_i^\diamond = \frac{1}{B}\sum_{l=1}^{B}\frac{\mathbb{1}_{\mathbf{X}_i^\diamond \in A_n(l)}}{N_{n,1}^\diamond(\mathbf{x};\Theta_l,\mathcal{D}_n)} =: \frac{1}{B}\sum_{l=1}^{B} Z_l, \tag{20}$$

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

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

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

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

Remark that

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

thus we have

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

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

Lemma 6 *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 as $W = u$ that fall into in $A_n(\Theta) = A_n(\mathbf{x};\Theta,\mathcal{D}_n)$ and $N_{n,u}^\diamond(A_n(\Theta)) = N_{n,u}^\diamond\left(\mathbf{x};\Theta,\mathbf{X}^{\diamond 1},\dots,\mathbf{X}^{\diamond n},\mathcal{D}_n\right)$, the number of observations of $\mathcal{D}_n^\diamond$ such as $W = u$ that fall into $A_n(\Theta)$. Then,*

$$\forall\varepsilon>0,\quad \mathbb{P}\left(\left|N_{n,u}(A_n(\Theta)) - N_{n,u}^\diamond(A_n(\Theta))\right| > \varepsilon\right) \leqslant 16(n+1)^{2d}e^{-\varepsilon^2/128n}.$$

Using Assumption 2 and Lemma 6, there exists C and M positive constants such that:

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

Thus since E_1^o converges to 0 in L^2 which implies that it also converges in L^1:

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

The term V_n can now be treated:

$$\begin{aligned}
V_n = & \left|\sum_{\substack{i=1\\W_i^{\diamond}=1}}^{n} \alpha_i^{\diamond}(\mathbf{x})\left[\mathbb{E}\left[Y^t(1)|\mathbf{X}_i^{\diamond}\right] - \mathbb{E}\left[Y^t(1)|\mathbf{X}=\mathbf{x}\right]\right]\right| \\
\leq & \left|\sum_{\substack{i=1\\W_i^{\diamond}=1}}^{n} \alpha_i^{\diamond}(\mathbf{x})\left[\mathbb{E}\left[Y^{new}(1)|\mathbf{X}_i^{\diamond}\right] - \mathbb{E}\left[Y^{new}(1)|\mathbf{X}=\mathbf{x}\right]\right]\right| \\
& + \left|\sum_{\substack{i=1\\W_i^{\diamond}=1}}^{n} \alpha_i^{\diamond}(\mathbf{x})\left[E_1^o(\mathcal{D},\mathbf{X}_i^{\diamond}) - E_1^o(\mathcal{D},\mathbf{x})\right]\right|.
\end{aligned}$$

We can state the following lemma from Lemma 2 in Meinshausen and Ridgeway (2006) and similar to Lemma 5 in Bénard et al. (2022).

Lemma 7 *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*

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

Combining the Lemma 7 with the continuity of τ_1, we get

$$\forall \ell \in [1,B], \forall \mathbf{x} \in \mathcal{X}, \sup_{\mathbf{z}\in A_n(\mathbf{x},\Theta_\ell,\mathcal{D}_n)} |\tau_1(\mathbf{z}) - \tau_1(\mathbf{x})| \xrightarrow[n\to\infty]{a.s.} 0. \tag{21}$$

Using this result we get that the first term tends to 0 almost surely:

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

Since each τ term is bounded, by dominated convergence theorem we have the L^1 convergence of this quantity.

The second term can be shown to be L^1 convergent to 0 using the same rationale than for U_n.

The quantity $\left|\hat{\tau}^{new}(\mathbf{x})-\tau^{new,\diamond}(\mathbf{x})\right|$ is now treated. We use the same decomposition and consider separately but in similar fashion $\left|\hat{\tau}_1^{new}(\mathbf{x})-\tau_1^{new,\diamond}(\mathbf{x})\right|$ and $\left|\hat{\tau}_0^{new}(\mathbf{x})-\tau_0^{new,\diamond}(\mathbf{x})\right|$:

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

Following the HTERF consistency proof, this term converges to 0 almost surely. Since all the Y terms are bounded, with dominated convergence theorem this term tends to 0 in L^1.

■

Appendix B. Generalisation bound

Using the proof in Section A, we get the following decomposition:

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

where

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

and

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

References

Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.

Clément Bénard, Sébastien Da Veiga, and Erwan Scornet. Mean decrease accuracy for random forests: inconsistency, and a practical solution via the sobol-mda. *Biometrika*, 109(4):881–900, 2022.

Ioana Bica and Mihaela van der Schaar. Transfer learning on heterogeneous feature spaces for treatment effects estimation. *Advances in Neural Information Processing Systems*, 35: 37184–37198, 2022.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19, 2006.

Chip Huyen. *Designing machine learning systems.* " O'Reilly Media, Inc.", 2022.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Bérénice-Alexia Jocteur, Véronique Maume-Deschamps, and Pierre Ribereau. Heterogeneous treatment effect-based random forest: Hterf. *Computational Statistics & Data Analysis*, page 107970, 2024.

Sören R Künzel, Bradly C Stadie, Nikita Vemuri, Varsha Ramakrishnan, Jasjeet S Sekhon, and Pieter Abbeel. Transfer learning for estimating causal effects using neural networks. *arXiv preprint arXiv:1808.07804*, 2018.

Nicolai Meinshausen and Greg Ridgeway. Quantile regression forests. *Journal of machine learning research*, 7(6), 2006.

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

Grace Wahba. 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, 2003.

Xuezhi Wang. *Active Transfer Learning*. PhD thesis, Ph. D. Dissertation. BAE Systems, 2016.

Xuezhi Wang and Jeff G Schneider. Generalization bounds for transfer learning under model shift. In *UAI*, pages 922–931, 2015.

Xuezhi Wang, Tzu-Kuo Huang, and Jeff Schneider. Active transfer learning under model shift. In *International Conference on Machine Learning*, pages 1305–1313. PMLR, 2014.

Song Wei, Hanyu Zhang, Ronald Moore, Rishikesan Kamaleswaran, and Yao Xie. Transfer learning for causal effect estimation, 2024.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International conference on machine learning*, pages 819–827. Pmlr, 2013.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International conference on machine learning*, pages 7523–7532. PMLR, 2019.