Erasmus University Rotterdam | Erasmus School of Economics

Master Thesis Econometrics and Management Sciences: Business Analytics and Quantitative Marketing

Evaluating Causal Machine Learning Methods

Name student: Christian Paul Wirths | ID number: 505921

Supervisor: Dr. Andrea A. Naghi

Second Assessor: Dr. Kathrin Gruber

Date final version: April 27, 2020

Abstract

Causal Machine Learning methods have lately added valuable contributions to the field of applied econometric research. Due to the wide amount of methods proposed, there is still lacking guidance on which methods to use in practise. Thus, in this study we revisit the 2016 Atlantic Causal Inference Competition by Dorie et al. (2019) and analyze the performance of methods which have not been directly compared, when estimating treatment effects of different aggregation levels in various complex simulation settings, where traditional methods would fail. Specifically, we employ Double Machine Learning (DML) to estimate the Average Treatment Effect (ATE), Generic Machine Learning for Heterogeneous Treatment Effects to estimate Group Average Treatment Effects (GATE) and Double Robust Modified Outcome Methods, Causal Forest as well as Bayesian Additive Regression Trees (BART) to estimate Individual Treatment Effects (ITE). Along the way, we retrieve all the higher-level estimates that can be derived from the more granular ones, for instance the ATE and GATE via the Causal Forest. Furthermore, we propose a new method by combining the predictive strengths of BART with the asymptotic results of the DML and the Generic framework. Consistently across all aggregation levels, the methods which model the response function most flexibly perform best. Additionally, we find that most of the Causal ML methods estimate the ATE well, however estimates become unstable with limited overlap. Furthermore, the methods are able to detect heterogeneity via automated algorithms, which we additionally demonstrate by analyzing a word experiment in the latest version of the General Social Survey (GSS).

Keywords: Causal Inference, Machine Learning, Causal Forest, Bayesian Additive Regression Trees, Double Robustness, Average/Heterogeneous Treatment Effect(s)

We thank SURFsara for access to the high-computing facilities of the Lisa Cluster.

The content of this thesis is the sole responsibility of the author and does not reflect the view of either Erasmus School of Economics or Erasmus University.

Contents

1 Introduction

1.1 The impact of Causal ML methods on economics

Quantifying treatment effects of an intervention and solving complex causal inference tasks are of great interest in several economic applications. For instance, policy makers might analyze the overall impact of an unemployment training on participants, by estimating the Average Treatment Effect (ATE). Additionally, if causal parameters vary with the participants' characteristics, it is insightful to investigate the treatment's heterogeneity. This is of high practical relevance, since program designers can find out which individuals benefit most from the program, or vice versa, on which participants the program might have a negative effect. The main estimation challenge is that the ground truth of the causal parameter is missing. For each data point, we are only observing the outcome given the treatment status, but we are lacking the potential counterfactual of the vice versa treatment. In randomized experiments, causal parameter can be derived by a fair comparison of the treatment and the control group. However, interaction terms describing heterogeneity have to be predefined ex-ante by the researcher, which can lead to overfitting or loss of valuable information (Chernozhukov, Demirer, Duflo, & Fernandez-Val, 2018b). When working with observational data, the treatment and control group can differ significantly. In this setting, one widely used identification strategy is the unconfoundedness assumption, stating that the potential outcome is independent of the treatment assignment, conditional on the covariates.

A considerable amount of approaches based on Machine Learning (ML) methods have provided striking contributions in this field of applied econometric research (Athey, 2018). Initially, ML methods are trained to predict an outcome in the test sample in an optimal way, by learning complex model specifications from the data via iterative algorithms. Furthermore, ML techniques can handle model specifications, where the amount of input parameters is exceeding the amount of observations in the data. Thus, they can flexibly capture heterogeneity along a large amount of covariates and dynamically learn confounding effects, without requiring the researcher to impose strong underlying assumptions or to specify the functional form of the model.

Lately, researchers have proposed several methods embedding these advantages within the estimation of causal parameters. To structure these methods, it helps to differentiate between general and specific solutions. General solutions provide a framework in which the researcher can plug in a ML method of choice, whereas the ML method used within a specific solution is pre-defined. One general approach to estimate ATE is Double/Debiased Machine Learning (DML) introduced by Chernozhukov et al. (2018a), where ML methods are predicting complex nuisance parameters in double robust score functions. DML provides asymptotically normal unbiased estimators, since the estimators converge at fast rates ($N^{-\frac{1}{2}}$) to the true parameter values, allowing to form valid confidence intervals.

To estimate heterogeneity in treatment effects, Chernozhukov et al. (2018b) set up a generic framework to conduct inference on Group Average Treatment Effects (GATE). They obtain consistent estimators based on the quantiles of the difference between the conditional mean outcome of treated and non-treated, both being predicted with ML methods. Knaus, Lechner, and Strittmatter (2018) show promising results in deriving point estimates of Individual Treatment Effects (ITEs), by modifying the outcome variable via the double robust score function also used in the DML, however the method is still lacking theoretical results for inference procedures. We refer to this approach as Double Robust Modified Outcome Method (DR MOM). In contrast, a specific solution to estimate heterogeneous treatment effects up until the individual level is the Causal Forest by Athey, Tibshirani, and Wager (2019). This approach builds up on traditional Random Forests by incorporating a splitting rule which maximizes heterogeneity, while keeping consistency and creating approximately normal distributed estimators. Lastly, a growing literature, including J. L. Hill (2011), J. Hill and Su (2013) and Green and Kern (2012) use Bayesian Additive Regression Trees (BART) to estimate causal effects by the posterior mean and intervals, following a Markov Chain Monte Carlo algorithm (MCMC).

1.2 Motivation for evaluating Causal ML methods

The above methods indicate that Machine Learning bears promising advances in the field of causal inference. However, most methods have been individually deployed in specific use cases or tested in simulation settings, where the data generating process might be tailored to the merits of the proposed approach. Moreover, due to the plethora of options available for the researcher, the literature is still lacking sufficient practical guidance on which method to use in which setting. Therefore, in this study, we will provide a comprehensive overview of selected ML methods for causal inference. Since randomized experiments can be costly, unethical or bear implementation errors, we will focus on observation data with non-random treatment assignment under the unconfoundeness assumption.

To accomplish that, we revisit the causal inference data analysis challenge, “Is Your SATT Where It’s At?”, which was part of the 2016 Atlantic Causal Inference Conference (Dorie et al., 2019). This competition has been specifically designed to neutrally compare causal inference methods. To maintain credibility, the researchers creating the data generating process were separate from the researchers submitting the methods. Furthermore, to imitate a real-life scenario as best as possible, the covariates are chosen from an observational study, while only the response and the treatment variable are being simulated. To test the methods across various DGPs, the authors simulate 77 highly complex scenarios with varying the degree of non-linearity in the *response* and *treatment function*, the *percentage of treated*, the *overlap* assumption (which indicates whether there is sufficient information to estimate the counterfactual across the entire covariate space), *alignment* (the level of correspondence between the response and treatment function) and the *heterogeneity* of the treatment effects (Dorie et al., 2019). Competition participants had the task to estimate the Average Treatment Effect on the Treated (ATT).

1.3 Research contribution

Dorie et al. (2019) made the DGP available open-source, encouraging further research. Firstly, we contribute to the literature by using the same simulation settings, but targeting other parameters of interest, specifically the ATE as well as heterogeneous effects, in form of GATEs and ITEs. More importantly, we focus on methods which were not used in the actual competition and which, to the best of our knowledge, have not been directly compared, but recently received high attention in the field of economics. Specifically, we estimate the ATE with the DML (Chernozhukov et al., 2018a), GATEs with the Generic Machine Learning for Heterogeneous Treatment Effects (Chernozhukov et al., 2018b), and ITEs with DR MOM (Knaus et al., 2018), as well as the Causal Forest (Athey et al., 2019). Along the way, we retrieve all the higher-level estimates that can be derived from the more granular ones, for instance the ATE and GATE of the Causal Forest from the ITE. Additionally, we estimate all target parameters of this research with one of the best performing methods of the competition by Dorie et al. (2019), BART MChains, to compare its performance with the introduced econometric Causal ML methods. Furthermore, due to its simplicity, a large number of applied economic papers use Ordinary Least Squares (OLS) to estimate the ATE (Słoczyński, 2015). Thus, we use OLS as a benchmark for traditional approaches and compare it with the new Causal ML methods. We evaluate the ATE and GATEs in terms of bias, RMSE, coverage and interval length and the ITEs in terms of the Precision in the Estimated Heterogeneous Effects (PEHE), which represents an averaged RMSE across simulation replications (J. L. Hill, 2011).

Secondly, we propose a new method by combining BART with the DML and Generic framework with the following idea. J. L. Hill (2011) and Dorie et al. (2019) use BART to derive causal effects from the posterior distribution. However, there are currently no results ensuring that the posterior mean is concentrated around the true mean function and that the MCMC sample is converging in polynomial time (Athey & Wager, 2019). Nevertheless, Chipman, George, McCulloch, et al. (2010) demonstrate that BART can have superior predictive performance compared to various statistical learning techniques including Boosting and Random Forest. To overcome the theoretical shortcomings of BART when estimating causal effects, but to utilize its strong predictive performance at the same time, we use BART to predict the nuisance parameters in the DML framework and the conditional expectation functions of the Generic Machine Learning for Heterogeneous Treatment Effects, as both approaches provide valid theoretical results. Aside from BART, following Chernozhukov et al. (2018a), we implement all general methods with Lasso, Single Regression Trees, Boosting, Random Forest and Neural Nets. Thereby, we contribute to the simulation studies of Knaus et al. (2018), who implement the DR MOM estimator only with Lasso and Random Forest and suggest adding other ML methods as further research. Within each general method, we specifically evaluate the performance of each ML method.

Thirdly, we compare the performance of the Causal ML methods when varying the functional form of the *response* and *treatment* model, *percentage of treated*, the *overlap* assumption, the treatment/response *alignment* and the level of *heterogeneity*, and thus stress-test their promised flexibility. Hereby, we contribute to the literature by disentangling the effect of each criteria separately, since Dorie et al. (2019) evaluate the performance mostly aggregated over all simulation settings.

Fourthly, to compare the Causal ML methods in practice, we apply them in an empirical application, by analyzing the public perception of the government's spending on *welfare* versus the spending on *assistance to the poor* based on the the most recent General Social Survey (GSS) data.

1.4 Outlook results

When estimating the ATE, we find that most of the Causal ML estimators perform reasonably well in terms of RMSE, however no method reaches nominal coverage across all simulation settings. The best performing method is BART MChains, followed by DML BART. These two methods reach nominal coverage only when there is full overlap, however the violation of overlap substantially deteriorates the performance of all methods. The high level of alignment and heterogeneity is also challenging, since all methods perform worse under these two criteria. Within the ATE analysis, all Causal ML estimators outperform OLS.

The best performing methods in estimating the most and least affected GATEs are BART MChains and Generic Boosting, where the former shows lower RMSEs and the latter higher coverage. Causal Forest performs well in terms of RMSE, but the coverage rates are remarkably low. Overall, the results of the estimation of the heterogeneous effects are worse than the ones from the ATE analysis, indicating that capturing heterogeneity is a more challenging task. When estimating ITEs, the best performing method is BART MChains, followed by DR MOM Boosting and DR MOM BART. Thus, across all aggregation levels of treatment effects, the Causal ML methods which model the response and the treatment function flexibly, by growing a sequence of trees based on a weak learner approach, perform best.

All Causal ML methods consistently detect Heterogeneous Treatment Effects in the empirical application, indicating that the perception of spending on *welfare* varies with the respondents' characteristics.

1.5 Structure of the research

This research is structured as follows. In Section 2, we provide a more detailed overview of the relevant literature on Machine Learning methods used for causal inference. Sections 3 covers the methodology, more specifically, the DML method, the DR MOM, the Generic Machine Learning for Heterogeneous Treatment Effects, the Causal Forest and Bayesian Additive Regression Trees. In Section 4, we present the data and evaluation criteria, based on the 2016 Atlantic Causal Inference Conference challenge. In Section 5, we present and discuss our results, by comparing the performance of the methods when estimating the ATE, GATEs and ITEs. Section 6 provides the results from the empirical application. Lastly, in Section 7, we draw conclusions and discuss topics for future research.

2 Literature review

Originally, various Machine Learning methods such as shrinkage approaches, Trees, Boosting, Random Forests or Neural Nets have been developed to solve challenging prediction problems (Friedman, Hastie, & Tibshirani, 2001). In a new and rapidly emerging literature, several researchers are proposing methods utilizing the strengths of Machine Learning to answer causal inference questions of treatment effects. A crucial identification strategy considered with these Causal ML methods is the unconfounded treatment assignment in observational studies[1].

The main challenge in estimating the Average Treatment Effect in observational studies is that we need to condition on a large set of covariates to receive a fair comparison between the test and the control group. A widely used method is the multiple linear regression, which includes several control variables to capture confounding effects (Gray et al., 1993). However, LaLonde (1986) demonstrated that traditional econometric estimates cannot fully replicate results from a randomized study. These findings initiated an ongoing debate on which methods are suitable when analyzing observational data. A considerable amount of these methods estimate the propensity score (reflecting the probability of being treated), the response function of the dependent variable, or both. For instance, Rosenbaum and Rubin (1983) estimate the propensity score to match a corresponding control group for the treated, or the propensity score is used to weight observations to create a pseudo-control population for the inferential group (Rosenbaum, 1987). Still, Hahn (1998) shows that the estimation of the ATE based on the propensity score is ancillary and can even cause a loss in efficiency, if the response function is estimated correctly. Methods combining both the propensity score and the response function seem to be most reliable or "double-robust", since Robins and Rotnitzky (1995) state that the estimator is consistent when either the propensity score or the response function is specified correctly. However, in most traditional techniques both the response and the propensity score are estimated using parametric models.

Chernozhukov et al. (2018a) build up on the idea of double robustness and introduce Double/Debiased Machine Learning (DML), which allows to estimate a low dimensional parameter of interest β_0 (e.g the ATE) in the presence of highly dimensional and complex nuisance parameters δ_0 (e.g the propensity score and the response function). In contrast to the traditional techniques, general ML methods are used to predict δ_0, which are then processed in moment equations to estimate β_0. The ML methods perform remarkably well in predictions, by trading-off over-fitting versus regularization bias. However, while variance is reduced due to regularization, a necessary bias is introduced. This bias causes the

[1] For a complete overview of other identification strategies which are out of the scope of this research, such as panel data settings or regression discontinuity design, see Section 4 of Athey (2018).

naive estimator (simply plugging in the ML-estimator of δ_0 in the equations of β_0) to be $N^{-1/2}$ inconsistent (Chernozhukov et al., 2018a). Therefore, the authors introduce two ingredients to keep the estimate root-N consistent: Firstly, they employ Neyman-orthogonal moments/ scores to estimate β_0 and secondly, they apply sample-splitting in the estimation process and keep efficiency via cross-fitting. The main contribution of their work is that the estimates are approximately unbiased and normally distributed, concentrated in the $N^{-1/2}$ neighborhood of the true parameter, allowing for valid confidence interval analysis (Chernozhukov et al., 2018a). DML estimates the ATE by using the score function of Robins and Rotnitzky (1995). To demonstrate the use in an empirical application, Chernozhukov et al. (2018a) use the DML to replicate a study which is measuring the effect of an insurance bonus on unemployment duration.

Analyzing heterogeneity in treatment effects is of equally high interest. In traditional econometric techniques, potential heterogeneous subgroups are specified ex-ante, by including interaction terms or by fitting separate models on pre-defined sample splits. However, this procedure has the disadvantage that valuable information is lost, if these subgroups are not chosen correctly. At the same time, simply increasing the amount of interaction terms or subgroups to overcome this limitation poses the risk of over-fitting. Therefore, Chernozhukov et al. (2018b) propose a strategy to draw inference on the key features of heterogeneous effects. Hereby, the Conditional Average Treatment Effect function is proxied using various Machine Learning methods. Based on these proxies, a linear prediction of the heterogeneous effects is derived, which evaluates whether heterogeneity is captured properly. Furthermore, Chernozhukov et al. (2018b) compute average effects sorted by impact groups, which are based on the quantiles of the HTE distribution. Due to repeated sample-splitting, the approach guarantees valid inference. This allows to test whether the group effects or the difference between the most and the least effected groups are statistically significant. Chernozhukov, Demirer, Duflo, and Fernandez-Val (2017) use the method to measure the impact of gender wage discrimination, demonstrating its potential use in observational studies.

Knaus et al. (2018) analyze the finite sample performance of Causal ML estimators for heterogeneous effects in several DGPs, by varying the size of the effects, the random noise level and the number of observations, concluding that the best performing group of estimators models the treatment assignment and the response function in multiple steps. One of these estimators is the DR MOM, which modifies the outcome with the double robust estimator by Robins and Rotnitzky (1995). Hereby, the authors use Lasso and Random Forest as Machine Learning methods to predict the response and treatment function, concluding that the Random Forest is superior.

As an alternative to estimate Heterogeneous Treatment Effects, Athey et al. (2019) propose a non-parametric estimation framework, Generalized Random Forest, which is inspired by the initial Random Forest of Breiman (2001). Hereby, the parameter of interest can be defined via the local moment conditions. In contrast to the standard Random Forest, the estimation follows an adaptive weighting mechanism based on trees using a splitting criterion, which maximizes heterogeneity (Athey et al., 2019). Since the iterative evaluation of the local moment equations within the splitting criterion is computationally very expensive, Athey et al. (2019) apply a computational trick and evaluate a gradient-based approximation, which makes the approach time-efficient. Furthermore, the method is based on honesty, meaning that a different data sample is used for building the tree than for the estimation (Athey & Imbens, 2016). As a main contribution, Athey et al. (2019) prove that the estimator is consistent and approximately Gaussian, so that the asymptotic variance and valid confidence intervals can be calculated. The Causal Forest is a specific case of the Generalized Random Forest, which combines several academic contributions, specifically the Causal Tree by Athey and Imbens (2016) and a pre-version of the Causal Forest by Wager and Athey (2018). For further performance improvement, Athey et al. (2019) added an additional orthogonalization step, as well as several tuning parameters inspired by the R-learner from Nie and Wager (2017), which denotes a regularized objective function targeting Heterogeneous Treatment Effects. The Causal Forest has been applied in various empirical applications, for instance to evaluate the impact of nudge-like interventions within high schools, as part of the National Study of Learning Mindsets (Athey & Wager, 2019).

Lastly, Chipman et al. (2010) propose Bayesian Additive Regression Trees (BART) to grow a sequence of trees to estimate conditional expectation functions by a weak learner, regularized by a prior. Sampling from the posterior is conducted via a back fitting MCMC algorithm. The author demonstrates that BART outperforms other highly tuned statistical learning techniques, such as Boosting and Random Forest, even with default parameters. J. L. Hill (2011) applies BART to model the outcome of the treated and non-treated, leading to the posterior distribution of Individual Treatment Effects, which can be used to calculate the causal effect on the population of interest. Furthermore, J. Hill and Su (2013) show that BART can identify areas of limited overlap by utilizing that the posterior standard deviation increases in these areas and the authors use this technique to measure the effect of breastfeeding on children's cognitive outcomes.

3 Methodology

3.1 Potential outcome framework and assumptions

Rubin (1974) defines the potential outcome framework as follows: For a set of i.i.d. data points with n observations, we observe a tuple (Y_i, X_i, D_i), where $Y_i \in \mathbb{R}$ is the outcome, $X_i \in \mathbb{R}^p$ is the covariates vector (control variables) and $D_i \in (0,1)$ is the treatment assignment. $D_i = 1$ represents the treatment group and $D_i = 0$ the control group. Then, let $Y_i(1)$ be the potential outcome if being treated and $Y_i(0)$ be the potential outcome if being non-treated. Since $Y_i = D_i Y_i(1) + (1 - D_i) Y_i(0)$, we only observe one of the two hypothetical outcomes. Thus, the Individual Treatment Effects (ITE) denoted by $\tau_i(X_i) = Y_i(1) - Y_i(0)$ are not directly observed. However, we can introduce identification strategies to derive treatment estimates. The Average Treatment Effect (ATE) is given by $\beta_0 = \mathbb{E}[Y_i(1) - Y_i(0)]$ (Imbens & Wooldridge, 2009). In applied econometric research, it is standard practice to use Ordinary Least Squares (OLS) to estimate

$$Y_i = \alpha_i + \beta_0 D_i + X_i \delta + \epsilon_i, \tag{1}$$

where $\hat{\beta}_0$ is interpreted as ATE (Słoczyński, 2015). To capture heterogeneity, the Conditional Average Treatment Effects (CATEs) are given by

$$\tau_0(x) = \mathbb{E}[Y_i(1) - Y_i(0)|X_i = x] = \mathbb{E}[Y_i(1)|X_i = x] - \mathbb{E}[Y_i(0)|X_i = x]. \tag{2}$$

Hereby, we can condition from a set of exogenous covariates up until the most granular level, representing the ITEs. Moreover, researcher are often interested in mixed aggregation levels such as the grouped effects based on the quantiles of the treatment distribution (Chernozhukov et al., 2018b). According to Knaus et al. (2018), GATEs are defined by

$$\tau_0(g) = \mathbb{E}[Y_i(1) - Y_i(0)|G_i = g] = \int \tau(x) f_{X_i|G_i=g}(x)dx. \tag{3}$$

As treatment assignment is not random in observational studies, we assume strongly ignorability of the treatment, consisting of two elements (J. L. Hill, 2011): Firstly, we impose the unconfoundedness assumption $Y_i(0), Y_i(0) \perp\!\!\!\perp D_i|X_i = x$, meaning that the potential outcome is independent of treatment assignment, conditional on the control variables. Secondly, we assume common support (also called full overlap)[2], given by $0 < Pr(D_i = 1|X_i = x) < 1$. This assumption ensures that the propensity score, i.e. the probability of being treated, is bounded away from zero and one, guaranteeing the existence of a hypothetical counterfactual for each treatment assignment. Combining these assumptions with Equation (2) yields

$$\tau_0(x) = \mathbb{E}[Y_i(1)|X_i = x] - \mathbb{E}[Y_i(0)|X_i = x] = \mathbb{E}[Y_i|D_i = 1, X_i = x] - \mathbb{E}[Y_i|D_i = 0, X_i = x], \tag{4}$$

implying that we can estimate the causal effects via conditional expectations functions (J. L. Hill, 2011). In the remaining of the research, we define the corresponding functions for the expected outcome given the treatment and control variables by $g_0(D_i, X_i) = E[Y_i|D_i = d, X_i = x]$, for the conditional mean regression marginalized over treatment by $l_0(x) = \mathbb{E}[Y_i|X_i = x]$ and for the propensity score by $m_0(x) = \mathbb{E}[D_i|X_i = x] = Pr(D_i = 1|X_i = x)$.

[2] Note that later on in this research we investigate the implications of violating this assumption.

3.2 Double/Debiased Machine Learning

3.2.1 General framework DML

The Double/Debiased Machine Learning (DML) framework by Chernozhukov et al. (2018a) provides valid point estimates and confidence intervals of a low dimensional parameter (β_0) in the presence of high dimensional nuisance parameters (δ_0). The main interest is in estimating the true value β_0 of the low-dimensional target parameter $\beta \in B$, where B is a non-empty measurable subset of $\mathbb{R}^{\mathrm{d}_\beta}$, with d_β being the dimension of β. Thereof, we introduce moment equations and assume that β_0 satisfies

$$E_p[\rho(W_i;\beta_0;\delta_0)] = 0, \tag{5}$$

where $\rho = (\rho_1, ...\rho_{d_\beta})'$ is a vector of defined score functions and W_i is a random element whose sample $(W_i)_{i=1}^N$ is available. δ_0 is the true value of the nuisance parameter $\delta \in T$, with T being a convex subset of some normed vector space.

As a key ingredient of the DML approach, δ_0 are estimated via non-parametric Machine Learning methods and are then processed in the moment equations to estimate β_0. To keep consistent estimates, we have to ensure that small deviations and errors in the nuisance function do not invalidate the moment conditions. For that purpose, we employ Neyman-orthogonality, which ensures that the moment conditions are locally insensitive to the value of the nuisance parameters (Chernozhukov et al., 2018a). Moreover, in the process of estimating β_0, we apply sample splitting to combat over-fitting, while keeping efficiency via cross-fitting. This is accomplished by estimating the nuisance parameters on a different data set from the one used to estimate the parameter of interest. Building up on the general moment estimation framework in Equation (5) and assuming Neyman-orthogonality, the implementation of DML including sample splitting and cross-fitting is described in Algorithm 1.

Algorithm 1: Data Splitting and Cross-Fitting in DML

```
Require: Let $[N] = \{1, ..., N\}$ be the set of all observations indices
Require: Define the number of $K$ folds
for each fold $k \in [K] = \{1, ..., K\}$ do
    Take a main sample by the random partition $I_k$ of [N] with size $n = N/K$
    Take the complementary auxiliary sample $I_k^c = [N] \setminus I_k$ with size $N - n$
    Use $I_k^c$ to estimate nuisance parameters $\hat{\delta}_{0,k} = \hat{\delta}_0((W_i)_{i \in I_k^c})$ with set of ML methods
end for
Plug in pooled ML estimates $\hat{\delta}_{0,k}$ to solve sample analog of Neyman-orthogonal moment equations
$\frac{1}{K}\sum_{k=1}^K \mathbb{E}_{n,k}[\rho(W_i;\beta_0;\hat{\delta}_{0,k})] = 0$ to estimate parameter of interest $\hat{\beta}_0$
```

For each fold k, the data is split randomly into the distinct main sample I_k and the complementary auxiliary sample I_k^c. The auxiliary sample is used to estimate the nuisance parameters, whereas the main sample is used to estimate the parameter of interest. The empirical applications of Chernozhukov et al. (2018a) use two folds ($K = 2$) and we will adapt this strategy.

Step 4 of Algorithm 1 reflects the estimation of the nuisance parameters via a ML method of choice. Following Chernozhukov et al. (2018a), we consider Lasso, Regression Trees, Random Forests, Boosting and Neural Nets. A detailed description of each ML method is given in Appendix A.6 and the input parameters are summarized in Appendix A.6.6 Table 17. Additionally, due to its strong predictive performance demonstrated by Chipman et al. (2010), we add BART to the DML framework.

Step 6 in Algorithm 1 outlines the pooling of $\hat{\delta}_{0,k}$, which is crucial to keep full efficiency of the estimator and to avoid information loss due to the sample splitting[3].

Lastly, if the moment equation in step 6 cannot be solved for zero, then $\hat{\beta}_0$ is defined as an approximate ϵ_N-solution of β_0, formally,

$$\parallel \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{n,k}[\rho(W_i;\hat{\beta}_0;\hat{\delta}_{0,k})] \parallel \leq \inf_{\beta\in B} \parallel \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{n,k}[\rho(W_i;\beta_0;\hat{\delta}_{0,k})] \parallel + \epsilon_N. \tag{6}$$

The main contribution of the DML is that it provides accurate point estimates, as well as valid inference, such as confidence intervals and p-values. Assuming linear score functions that are Neyman-orthogonal and nuisance parameter estimates that belong to the nuisance realization set $T_N \subset T$, Chernozhukov et al. (2018a) derive that the estimator $\hat{\beta}_0$ is approximately linear and centered Gaussian. Furthermore, the variance estimator of the point estimate is given by

$$\hat{\sigma}^2 = \hat{J}_0^{-1}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{n,k}[\rho(W_i,\hat{\beta}_0,\hat{\delta}_{0,k})\rho(W,\hat{\beta}_0,\hat{\delta}_{0,k})']\hat{J}_0^{-1\prime}, \tag{7}$$

where $\hat{J}_0$ are score functions only depending on δ_0.

The final step in the DML approach is to repeat Algorithm 1 over S splits. According to Chernozhukov et al. (2018a), this makes the results more robust, since it accounts for the randomization of the sample splitting, and they propose 100 splits in empirical applications. As we implement the DML in 77 different simulation settings, each consisting of 100 replications, comprising of a total of 7700 datasets, we only consider 10 splits per replication, which we believe is sufficient to receive robust results. The final DML estimates are the median value of the point estimate, the median of the standard errors and modified standard errors incorporating the uncertainty of sample splitting

$$\hat{\beta}_0^m = med\{\hat{\beta}_{0,s}\},\ \hat{\sigma}_0^{2,m} = med\{\hat{\sigma}_{0,s}^2\}_{s=1}^S,\ \hat{\sigma}_0^{2,m*} = med\{\hat{\sigma}_{0,s}^2 + ((\hat{\beta}_{0,s} - \hat{\beta}_{0,s}^m)(\hat{\beta}_{0,s} - \hat{\beta}_{0,s}^m)'\}_{s=1}^S. \tag{8}$$

3.2.2 Moment equations used in the DML approach

In the following, we elaborate the specification and the Neyman-orthogonal score function, which lead to the estimation of the ATE, given the potential outcome framework under unconfoundedness

[3] As an alternative, Chernozhukov et al. (2018a) propose to solve the moment equation for each k-fold separately to obtain $\hat{\beta}_{0,k}(I_k, I_k^c)\ \forall\ k \in [K]$ and estimate $\hat{\beta}_0$ by taking the average.

assumption[4]. Chernozhukov et al. (2018a) propose the interactive model denoted by the binary $D_i \in \{0,1\}$ given by

$$Y_i = g_0(D_i, X_i) + U_i \qquad \mathbb{E}[U_i|X_i, D_i] = 0, \tag{9}$$

$$D_i = m_0(X_i) + V_i \qquad \mathbb{E}[V_i|X_i] = 0. \tag{10}$$

Hereby, the confounding effects influence the outcome variable through $g_0(D_i, X_i)$ and the treatment through the propensity score $m_0(X_i)$. In traditional approaches, both of these highly complex functions have to be pre-specified by the researcher, but DML bears the advantage of dynamically modelling these terms via Machine Learning. Robins and Rotnitzky (1995) set up a double robust estimator by

$$Y^*_{i,DR}(W_i, \delta_0) = g_0(1, X_i) - g_0(0, X_i) + \frac{D_i(Y_i - g_0(1, X_i))}{m_0(X_i)} - \frac{(1 - D_i)(Y_i - g_0(0, X_i))}{1 - m_0(X_i)}. \tag{11}$$

Since the ATE is given by $\beta_0 = \mathbb{E}[g_0(1, X_i) - g_0(0, X_i)]$, Chernozhukov et al. (2018a) introduce the moment equations

$$\rho(W_i; \beta_0; \delta_0) = Y^*_i{}_{DR}(W_i, \delta_0) - \beta_0. \tag{12}$$

Solving the sample analog expression shows that $\hat{\beta}_0$ is equal to the mean over all observations denoted by $\hat{\beta}_0 = \frac{1}{N}\sum_{i=1}^{N} Y^*_i{}_{DR}(W_i, \hat{\delta}_0^{(-i)})$, where superscript $^{(-i)}$ denotes that the nuisance parameter are estimated via ML-methods on the complementary data sets, see Equation (36) in Appendix A.8. Furthermore, according to Chernozhukov, Chetverikov, et al. (2017), an approximate standard error for this estimator is given by standard deviation $\hat{\sigma}$ over all observations divided by $\sqrt{N}$, where $\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N} \hat{\rho}_i{}^2 = \frac{1}{N}\sum_{i=1}^{N}(Y^*_i{}_{DR}(W_i, \hat{\delta}_0^{(-i)}) - \hat{\beta}_0)^2$. Within the calculation of the estimate, we follow the practical advice of Chernozhukov et al. (2018a) to trim propensity scores at the cutoff points of 0.01 and 0.99 to diminish the disproportionate impact of extreme propensity score weights.

Knaus et al. (2018) show that the double robust estimator by Robins and Rotnitzky (1995) can also be used to derive point estimates of the Conditional Average Treatment Effects up until the individual level, since $E[Y^*_{i,DR}|X_i = x] = \tau_0(x)$. Similar to the DML, we use the plug-ins of the cross-fitted estimated nuisance parameters to compute Equation (11). However, instead of taking the mean of the expression, we use it as a modified outcome to regress $Y^*_{i,DR}$ on X_i, yielding $\hat{\tau}_0^{DR}(x)$. We refer to this approach as the Double Robust Modified Moment Method (DR MOM). Note that in contrast to the DML approach, we do not repeat the method over S splits and no asymptotic theory is derived for this estimate.

[4] Depending on the score function, the DML framework can also target several other causal parameters which are not in scope of this research, such as partial linear regression models, IV-estimation and Local Average Treatment Effects (LATE), see Chernozhukov et al. (2018a) for more details on the corresponding moment equations.

3.3 Generic Machine Learning for Heterogeneous Treatment Effects

Learning the CATE function $\tau_0(x)$ with general ML methods by taking the difference of the two regressions in Equation (2) has the limitation that there is still no uniformly valid inference available. Therefore, Chernozhukov et al. (2018b) consider a radically different approach. They propose a method using ML proxies for the estimation and inference on the key features of $\tau_0(x)$, rather than $\tau_0(x)$ itself. The key features of interest in this research are the Best Linear Predictor (BLP) to estiamte ATE, and the GATEs defined by the average effects sorted by impact groups (Chernozhukov et al., 2018b). Given the potential outcome framework, we consider the regression function

$$Y_i = g_0(0, X_i) + D_i \tau_0(X_i) \qquad E_p[U_i|X_i, D_i] = 0, \tag{13}$$

where $g_0(0, X_i) = E[Y_i|D_i = 0, X_i]$ and $\tau_0(X_i) = E[Y_i|D_i = 1, X_i] - E[Y_i|D_i = 0, X_i]$. Chernozhukov et al. (2018b) deploy the method in randomized experiments, where due to random treatment assignment, the decomposition of $\tau_0(X_i)$ is true by definition. Nevertheless, as shown in Equation (4), assuming strong ignorability, whereby we have zero unobserved bias and we have captured all confounding effects as well as common support, it also holds potential in observational studies to condition flexibly on very high dimensional controls (Chernozhukov, Demirer, et al., 2017). As an additional step in observational studies, we also estimate the propensity score $m_0(X_i)$, since it is no longer constant. We emphasize that in this setting, the estimates reflect the best approximation of the causal effects.

To obtain valid inference for these key features, we apply sample splitting. We split the data randomly into two distinct sets, the auxiliary sample $\text{Data}_A = (Y_i, D_i, X_i)_{i \in A}$ and the main sample $\text{Data}_M = (Y_i, D_i, X_i)_{i \in M}$. Using ML methods, we train $G(0, x)$ as a proxy for $g_0(0, x)$ and $T(x)$ as a proxy for $\tau_0(x)$ on the auxiliary sample A. According to Chernozhukov et al. (2018b), the resulting estimates can even be inconsistent, since the parameters are post-processed as proxies within the specifications of the key features in the main sample M.

Firstly, to retrieve the best linear predictor we set up the weighted linear projection

$$Y_i = \alpha' C_{1i} + \beta_1(D_i - m_0(X_i)) + \beta_2(D_i - m_0(X_i))(T_i - \mathbb{E}\,T_i) + e_i \quad \mathbb{E}[w(X_i)e_i C_i] = 0 \quad i \in M, \tag{14}$$

where $T_i := T(X_i)$, $w(X_i) = \{m_0(X_i)(1 - m_0(X_i))\}^{-1}$, $C_i := (C_{1i}, C_{2i})$, $C_{1i} := C_1(X_i)$, such as $C_{1i} = (1, G(0, X_i))$ and $C_{2i} := [D_i - m_0(X_i), (D_i - m_0(X_i)(T(X_i) - (T_i - \mathbb{E}\,T_i))]$. Hereby, the interaction $(D_i - m_0(X_i))(T_i - ET_i)$ weighted by $w(X_i)$ is orthogonal to $(D_i - m_0(X_i))$, which ensures that the strategy works under misspecifications in very high-dimensional problems similar to the DML approach (Chernozhukov et al., 2018b). As a main result, the best linear predictor is defined by

$$\beta_0 + \beta_1(T(X_i) - \mathbb{E}\,T_i) = \text{BLP}[\tau_0(X_i)|T(X_i)] \tag{15}$$

and can be estimated using weighted OLS. $\beta_0 = \mathbb{E}\,\tau_0(X_i)$ represents the ATE, and $\beta_1 = \frac{\text{Cov}(\tau_0(X_i),T(X_i))}{Var(T(X_i))}$ indicates the heterogeneity level. According to Chernozhukov et al. (2018b), if $T(X_i)$ is a perfect proxy for $\tau_0(X_i)$, then $\beta_1 = 1$. In contrast, if there is no heterogeneity, then $\beta_1 = 0$. Generally, rejecting the hypothesis $\beta_1 = 0$ indicates that there is relevant heterogeneity. Here, it should be noted that Athey et al. (2019) incorporate this test statistic to evaluate the heterogeneity within the Causal Forest.

Secondly, the approach allows to estimate sorted Group Average Treatment Effects (GATEs) $E[\tau_0(X_i)|G_k]$ for groups $k =, 1..., K$. To explain variation in treatment effects, the groups can be assigned based on non-overlapping quantiles, where G_k denotes the group membership to the k-quantile of the distribution of the heterogeneous effects. If $T(X_i)$ is consistent for $\tau_0(X_i)$, we can introduce the monotonicity assumption $\mathbb{E}[\tau_0(X_i)|G_1] \leqslant ... \leqslant \mathbb{E}[\tau_0(X_i)|G_K]$, which holds asymptotically (Chernozhukov et al., 2018b). The GATEs can be recovered by the weighted linear projection

$$Y_i = \alpha' C_{1i} + \sum_{k=1}^{K} \phi_k * (D_i - m_0(X_i)) * 1(G_k) + v_i \quad \mathbb{E}[w(X_i) v_i W_i] = 0 \quad i \in M, \tag{16}$$

where $W_i = (C'_{1i}, W'_{2i})'$ and $W_{2i} = (\{D_i - m_0(X_i))1(G_k)\}_{k=1}^{K})$'. Comparable to the BLP, the interaction $(D_i - m_0(X_i)) * 1(G_k)$ is orthogonal to all regressors depending on X_i. Thereby, the projection coefficients ϕ_k represent the GATEs parameters

$$\phi = (\phi)_{k=1}^{K} = (E[\tau_0(X_i)|G_k])_{k=1}^{K}. \tag{17}$$

To target inference, we can investigate single grouped effects by testing $\phi_k = 0$. Furthermore, we can analyze heterogeneity by comparing different groups such as testing $\phi_k - \phi_j = 0$ for $k \neq j$.

The method bears two sources of uncertainty. On the one hand, we have the estimation uncertainty regarding the parameter of interest θ (such as GATEs), conditional on the data split. On the other hand, we have the uncertainty induced by data splitting. Therefore, we will repeat the approach over $1, ..., B$ repetitions. Each step introduces a different random split of the auxiliary sample and the main sample. Thereby, we create estimates that are more robust than the ones from a single random split. All parameters implicitly depend on the auxiliary sample, which we denote by $\theta = \theta_A$. Chernozhukov et al. (2018a) propose to adjust the point estimates by the median over all B repetitions $\hat{\theta} := \text{Med}[\hat{\theta_A}|\text{Data}]$. With significance level α and confidence interval (CI) of length $1-2\alpha$, the adjusted CI are denoted by $[l, u] := [\overline{\text{Med}}|L_A, \underline{\text{Med}}|U_A]$, where $\underline{\text{Med}}$ is the lower and $\overline{\text{Med}}$ the upper definition of the median. Furthermore, Chernozhukov et al. (2018b) show that the adjusted p-values have the form $P(p_A \leqslant \frac{\alpha}{2}|\text{Data} \geq 1/2)$. More specifically, it holds that the p-value is at largest $\frac{\alpha}{2}$, for at least 50% of the random splits. Small values provide evidence against the corresponding null hypothesis of interest (Chernozhukov et al., 2018b). Algorithm 2 summarizes the aforementioned steps.

In empirical studies, Chernozhukov et al. (2018b) choose 100 repetitions. Since we will apply the approach in a simulation study with 100 replications, we choose 10 repetitions per replication. Similar

Algorithm 2: Implementation of Generic Machine Learning to calculate BLP and GATE

```
Require: Let $[N] = \{1, ..., N\}$ be the set of all observations indices
Require: Define the number of $B$ repetitions
Require: Compute the propensity scores for $m_0(X_i)$ $\forall$ i $\in$ N
1: for $b \in [B] = \{1, ..., B\}$ do
2:     Split the data randomly into auxiliary sample Data$_A$ and main sample Data$_M$
3:     Train $Y_i(0) = G_0(D_i = 0, X_i)$ via ML methods $\forall$ i $\in \{A \cup D_i = 0\}$
4:     Train $Y_i(1) = G_1(D_i = 1, X_i)$ via ML methods $\forall$ i $\in \{A \cup D_i = 1\}$
5:     Predict $\hat{Y}_i(1)$ and $\hat{Y}_i(0)$ i $\in$ M, save predicted baseline effect $\hat{G}_0(X_i) = \hat{Y}_i(0)$ and predicted
       treatment effect $\hat{T}(X_i) = \hat{Y}_i(1) - \hat{Y}_i(0)$
6:     Calculate BLP and GATE according to Equations (14) and (17). BLP includes ATE.
7: end for
8: Take the medians and retrieve adjusted point estimates, confidence intervals and p-values
```

to the DML, in steps 3-5, we choose Lasso, Trees, Forest, Boosting and Neural Nets (described in Appendix A.6 with input parameters summarized in Appendix Table 17) and BART as ML methods. Furthermore, we also reduce the disproportionate impact of extreme propensity score by trimming the predicted propensity scores outside the cutoff of 0.01 and 0.99 (Chernozhukov, Demirer, et al., 2017).

3.4 Generalized Random Forest

3.4.1 General framework

Athey et al. (2019) introduce the Generalized Random Forest, a flexible method to use the Random Forest established by Breiman (2001) (see Appendix A.6.4) for estimating any quantity of interest identified by the local moment conditions, while accounting for the presence of nuisance parameters and providing valid statistical inference. Considering the corresponding moment equations in the potential outcome framework under the unconfoundedness assumption (Section 3.1), this framework can be used to estimate Conditional Average Treatment Effects via the so-called Causal Forest. In a nutshell, the Causal Forest composes of tree estimates grown on random subsets of the data, where each tree is identifying Heterogeneous Treatment Effects. The frameworks builds up on many properties of the Random Forest introduced by Breiman (2001), such as recursive partitioning, subsampling and random split selection. In addition, there are four fundamental ingredients of the Causal Forest: Firstly, the forest predictors are not derived by averaging tree estimates, but rather by an adaptive nearest neighborhood mechanism. Secondly, to reduce bias in tree predictors and to allow valid inference, the Causal Forest follows an honest approach, where one sample is used to construct the partition and the other sample to estimate the parameter (Wager & Athey, 2018). Thirdly, the objective of the split criterion within each tree is not to minimize a predictive error, but instead to capture heterogeneity in the parameter of interest. Fourthly, a computationally efficient gradient based approximation of the estimation equation is introduced, since evaluation rules are re-considered many times for each split in each tree. We elaborate on each of these points in turn.

Adopting the notation of Athey et al. (2019), the implementation of the Generalized Random Forest works as follows. We consider n independent and identically distributed random samples $i = 1...n$. In each sample, we observe quantity O_i and auxiliary covariates X_i. The goal is to estimate the parameter of interest $\beta_0(x)$ in the presence of nuisance parameters $\delta_0(x)$, by solving the local estimation equations based on the score function $\rho(\cdot)$

$$\mathbb{E}[\rho_{\beta_0(x),\delta_0(x)}(O_i)|X_i = x] = 0 \quad \forall x \in X, \tag{18}$$

where $x \in \mathrm{X}$ specifies a test point of covariates, for which we aim to estimate the causal effect. According to Athey et al. (2019), computing the forest predictors by averaging the tree-based predictors, such as in the Breitman's forest, would incorporate a bias when including noisy moment solutions as in Equation (18). To avert this issue, we introduce similarity weights α_i, reflecting the relevance of the i-th training example to the estimation of $\beta_0(\cdot)$ at test point x (Athey et al., 2019). Following the approach of Hothorn, Lausen, Benner, and Radespiel-Tröger (2004), we first build a forest consisting of B trees indexed by $b = 1, ..., B$. For each b tree, we denote the neighborhood $L_b(x)$ as the set of training examples belonging to the same "leaf" as x. Then, the weights α_i represent the frequency with which the i-th training example falls into the same leaf as x, given by

$$\alpha_{bi}(x) = \frac{1(\{X_i \in L_b(x)\})}{|L_b(x)|}, \quad \alpha_i(x) = \frac{1}{B}\sum_{b=1}^{B} \alpha_{bi}(x). \tag{19}$$

Including these weights, we solve for the $\hat{\beta}(x)$ and $\hat{\delta}(x)$ in Equation (18) by minimizing the sample analog

$$(\hat{\beta}(x), \hat{\delta}(x)) \in \text{argmin}_{\beta_0,\delta_0}\{||\sum_{i=1}^{n} \alpha_i(x)\rho(O_i; \beta_0(x), \delta_0(x)||_2\}, \tag{20}$$

which simplifies to $\sum_{i=1}^{n} \alpha_i(x)\rho(O_i; \beta(x), \delta(x)) = 0$, in case the expression contains a unique root (Athey et al., 2019).

To combat over-fitting and high variance of single trees, we consider a random subsample I of size s from the full sample S for each tree b. Additionally, following Wager and Athey (2018), we anticipate honest estimation, by building trees and estimating the weights on two distinct sets. On that note, we split subsample I into two distinct sets J_1 and J_2, where J_1 is used to build the trees and J_2 to retrieve the weights α_i. To provide an overview, Algorithm 3 describes the Generalized Random Forest framework, where steps 5 - 10 denote the honestly approach (Athey et al., 2019).

When building each tree in step 6, the objective of the splitting criterion is tailored to maximize heterogeneity in the following way: Given the sample J_1, every splits starts with a parent node $P \subseteq X$.

Algorithm 3: Generalized Random Forest by Athey et al. (2019) with honesty and sub-sampling

```
Require: Fix tuning parameters or specify which ones should be tuned via R-learner
Require: Set $B$ number of trees
function generalizedrandomforest(set of examples $S$, test point $x$)
Initialize weight vector $\alpha$ by vector of zeros of length $|S|$:   $\alpha \leftarrow zeros(|S|)$
for $b \in [B] = \{1, ..., B\}$ do
    Draw a subsample $I$ of size $s$ from $S$: $I \leftarrow subsample(S, s)$
    Split subsample I into two non-overlapping sets: $J_1, J_2 \leftarrow splitsample(I)$
    Use $J_1$ to build tree: $T \leftarrow gradienttree(J_1, X)$
    Return elements of $J_2$ that fall into the same leaf as $x$ in $T$: N $\leftarrow neighbors(x, T, J_2)$
    for $\forall e \in N$ do
        Update weights: $\alpha[e] += \frac{1}{|N|}$
    end for
end for
Solve Equation (20) for $\hat{\beta}_0(x)$ with weights $\frac{\alpha}{B}$
```

At this parent node, the solution to the estimation equation is denoted by

$$(\hat{\beta}_P, \hat{\delta}_P)(J_1) \in \operatorname{argmin}_{\beta,\delta}\Big\{\Big|\Big| \sum_{\{i \in J_1 : X_i \in P\}} \rho_{\beta,\delta}(O_i)\Big|\Big|_2\Big\}. \tag{21}$$

In the subsequent notations, the dependency on J_1 is omitted when being implicit. As a next step, the parent node P is divided into two children C_1, $C_2 \subseteq X$ in a recursive manner. Hereby, we perform a greedy search to find these splits, which capture the heterogeneity of $\beta_0(x)$ as best as possible (Athey et al., 2019). Formally, we aim to maximize the criterion

$$\Delta(C_1, C_2) := \frac{n_{C_1} n_{C_2}}{n_P^2}\big(\hat{\beta}_{C1}(J_1) - \hat{\beta}_{C2}(J_1)\big)^2, \tag{22}$$

where $\hat{\beta}_{C1}, \hat{\beta}_{C2}$ denote the solution to Equation (21) at the two children, and n_{C_1}, n_{C_2}, n_p are the number of observations at the children and parent nodes, respectively.

Evaluating Criterion (22) and solving the corresponding moment equations over all possible splits is computationally quite demanding. Thus, Athey et al. (2019) propose to optimize an approximate criterion $\widetilde{\Delta}(C_1, C_2)$ based on gradient based approximations of $\hat{\beta}_{C_1}$ and $\hat{\beta}_{C_2}$. Formally, we can define the gradient based approximations by

$$\widetilde{\beta_C} = \widetilde{\beta_P} - \frac{1}{|\{i : X_i \in C\}|} \sum_{\{i : X_i \in C\}} \zeta^T A_p^{-1} \rho_{\hat{\beta}_P, \hat{\delta}_P}(O_i), \tag{23}$$

where $\hat{\beta}_p, \hat{\delta}_p$ are the estimators at the parent node resulting from Equation (21) and ζ is a vector consisting of the β coordinate from (β, δ). A_p is a consistent estimate for the gradient of the score function $\nabla \mathbb{E}[\rho_{\hat{\beta}_P,\hat{\delta}_P}(O_i)|X_i \in P]$ denoted by

$$A_p = \frac{1}{|\{i : X_i \in P\}|} \sum_{\{i:X_i \in P\}} \nabla \rho_{\hat{\beta}_P,\hat{\delta}_P}(O_i). \tag{24}$$

Given this setup, the extensive recursive partitioning to obtain optimal splits simplifies to a labeling and a regression step (Athey et al., 2019). Firstly, in the labeling step we compute pseudo outcomes o_i given by

$$o_i = -\zeta^T A_p^{-1} \rho_{\hat{\beta}_p,\hat{\delta}_p}(O_i) \in \mathbb{R}, \tag{25}$$

where $\hat{\beta}_p, \hat{\delta}_p, A_p$ have been previously estimated at the parent node. Secondly, in the regression step, we are deploying a standard tree on the pseudo outcomes to split P into C_1 and C_2. Thereby, the approximate criterion is given by

$$\widetilde{\Delta}(C_1, C_2) = \sum_{j=1}^{2} \frac{1}{|\{i : X_i \in C_j\}|} \Big(\sum_{\{i:X_i \in C_j\}} o_i \Big)^2, \tag{26}$$

which we aim to maximize when accessing children split possibilities. According to Athey et al. (2019), the main benefit of this approximate criterion is that given a feature, we can evaluate all possible split points with a unique pass over the data, instead of optimizing Criterion (22) for each split separately. We provide an overview of the tree estimation step within the Generalized Random Forest framework in Algorithm 4 in Appendix A.9.

3.4.2 Asymptotic analysis of Generalized Random Forests

Wager and Athey (2018) have derived valid asymptotic theory for predictors resulting from the honest tree averages, based on U-statistics. Since the Generalized Random Forest estimates $\hat{\beta}_0$ are based on weighted predictors, Athey et al. (2019) transfer the theoretical results from Wager and Athey (2018) on a pseudo-forest estimate $\beta_0^*(x)$, which is representative for $\hat{\beta}_0$, showing that both $\beta_0^*(x)$ and $\hat{\beta}_0$ are asymptotically normal. Formally, Athey et al. (2019) derive that under weak assumptions, there is a sequence $\sigma_n(x)$ for which $\frac{\hat{\beta}_n(x)-\beta(x)}{\sigma_n(x)} \sim N(0,1)$.

Furthermore, σ_n can be used to construct asymptotically valid Gaussian confidence intervals for $\beta(x)$ on $\hat{\beta}(x)$, since $lim_{n\to\infty} \mathbb{E}[\beta(x) \in (\hat{\beta}_n(x) \pm \Phi^{-1}(1-\frac{\alpha}{2})\hat{\sigma}_n(x))] = \alpha$ (Athey et al., 2019). An estimator for $\hat{\sigma}_n^2(x)$ can be derived using the Delta-method with

$$\hat{\sigma}_n^2(x) := \zeta^T \hat{V}(x)^{-1} \hat{H}_n(x) (\hat{V}_n(x)^{-1})^T \zeta. \tag{27}$$

While $\hat{V}(x)$ denotes a specific curvature parameter which is not directly linked to the forest based methods, the inner variance $H_n(x; \beta_0(x), \delta_0(x))$ depends on the true parameters of $\beta_0(x)$ and $\delta_0(x)$, which, however, are not directly accessible (Athey et al., 2019). As an alternative to estimate $\hat{H}_n$, the so-called bootstrap of little bags, or the half-sampling estimator by Sexton and Laake (2009) is considered. For that purpose, we need to modify the sub-sampling step. Given an integer $l \geq 2$, we draw $g = 1, ..., B/l$ random half-samples H_g of size $\frac{n}{2}$. Then, for each tree b in $b = 1, ..., B$ we use the sub-samples I_b, such that $I_b \subseteq H_{[b/l]}$ (Athey et al., 2019). Intuitively, a small group of trees in trained on each half-sample. According to Athey et al. (2019), this has the advantage that we can identify $\hat{H}_n$ by the composition of the between group and within group variation. Formally,

$$\mathbb{E}_{ss}\,[(\frac{1}{l}\sum_{b=1}^{l}\rho_b - \rho)^2] = \hat{H}_n(x) + \frac{1}{1-l}\,\mathbb{E}_{ss}\,[\frac{1}{l}\sum_{b=1}^{l}(\rho_b - \frac{1}{l}\sum_{b=1}^{l}\rho_b)^2]. \tag{28}$$

For the practical implementation, we should consider that the larger the l, the larger the amount of trees B should be, in order to obtain valid confidence intervals.

3.4.3 Causal Forest implementation for Heterogeneous Treatment Effects

Under the potential outcome framework from Section 3.1, we need to define the model specifications and the corresponding score function to estimate Heterogeneous Treatment Effects with the Generalized Random Forest framework, also called Causal Forest. Thereof, according to Athey and Wager (2019), the Causal Forest integrates the R-learner objective function for Heterogeneous Treatment Effect estimation by Nie and Wager (2017), denoted by

$$\tau(\cdot) = \text{argmin}_\tau\{\sum_{i=1}^{n}((Y_i - \hat{l}_0^{-i}(X_i)) - \tau(X_i)(D_i - \hat{m}_0^{(-i)}(X_i)))^2 + \lambda_n(\tau(\cdot))\}, \tag{29}$$

where $l_0(x) = \mathbb{E}[Y_i|X_i = x]$ and $m_0(x) = \mathbb{E}[D_i|X_i = x]$ are the conditional marginal expectations of Y_i and D_i. $\lambda_n(\tau(\cdot))$ denotes a regularization parameter to control the complexity of $\tau(\cdot)$ and the superscripts$^{(-i)}$ indicate the 'out-of-bag' predictions, meaning the marginal expectations are estimated without the i-th observation. The later is introduced because Athey et al. (2019) demonstrate that an initial orthogonalization step by considering residuals estimated on a different data sample (similar to the DML method, see Section 3.2.1) yields more robust results than versions of the Causal Forest without this feature. Thus, we firstly fit two separate regressions forests to estimate $\hat{l}_0(\cdot)$ and $\hat{m}_0(\cdot)$ and then we grow a Causal Forest by solving

$$\hat{\tau}(x) = \frac{\sum_{i=1}^{n}\alpha_i(x)(D_i - \hat{m}_0^{-i}(X_i)(Y_i - \hat{l}_0^{-i}(X_i))}{\sum_{i=1}^{n}\alpha_i(x)(D_i - \hat{l}_0^{(-i)}(X_i))^2}, \tag{30}$$

where we implement the adaptive neighborhood weighting, as defined in Equation (19) (Athey & Wager, 2019). Hereby, we use the minimization of the R-objective in Equation (29) to tune the parameters of the Causal Forest via cross-validation. The standard tuning parameters are the sample fraction, the number of variables considered at each split and the minimum number of observations in terminal nodes. Additionally, we tune Causal Forest specific parameters such as the honesty fraction, which indicates the share used to build the tree and to make predictions, and a penalty term, which evaluates how imbalanced the splits are, when maximizing the heterogeneity splitting criterion. Moreover, we define the number of trees by $B = 8000$ and $l = 2$, which is required to define the set of half-samples to compute confidence intervals. Lastly, when estimating the ATE with the Causal Forest, we use the weighted estimator $\hat{\beta}_{ATE} = \sum_{i=1}^{n} m(X_i)(1 - m(X_i))E[Y(1) - Y(0)|X_i = x]/\sum_{i=1}^{n} m(X_i)(1 - m(X_i))$ where $m(x) = P[D_i = 1|X_i = x]$ proposed by Li, Morgan, and Zaslavsky (2018), which is recommended in case of poor overlap.

3.5 Bayesian Additive Regression Trees

3.5.1 BART set-up and prior specification

Bayesian Additive Regression Trees (BART) by Chipman et al. (2010), like Random Forest and Boosting, belong to the group of ensemble methods, since they include a sum of trees to approximate the function $f(x) = E[Y|X]$. Hereby, BART grows a sequence of trees based on a weak learner approach, yet in contrast to Boosting, each tree is weakened by a regularization prior (Chipman et al., 2010). Intuitively, this prior constraints the contribution of each tree, such that each tree explains a different minor portion of $f(x)$. Given $j = 1, ..., m$ trees, where $M_j = \{\mu_1, \mu_2, ..., \mu_b\}$ denotes the parameter set of the b terminal nodes in each tree T_j, Chipman et al. (2010) denote the sum of trees model by

$$Y = (\sum_{j=1}^{m} g(x, T_j, M_j)) + \epsilon, \quad \epsilon \sim N(0, \sigma^2). \tag{31}$$

BART introduces a prior on (T_j, M_j) $\forall j = 1, ..., m$ and σ. Furthermore, the authors show that the prior can be divided into the independent components

$$p((T_1, M_1), ..., (T_m, M_m), \sigma) = [\prod_j p(M_j|T_j)p(T_j)]p(\sigma), \tag{32}$$

where $p(M_j|T_j) = \prod_i p(\mu_{ij}|T_j)$. Thus, we can simplify the prior specification by choosing a prior for the tree structure $p(T_j)$, a prior for the values in the terminal nodes conditional on the tree structure $p(\mu_{i,j}|T_j)$ and a prior for the noise of the residuals' standard deviation $p(\sigma)$. We choose robust default prior parameters, which are partially derived by a data-driven approach and which show a competitive performance compared to dynamic priors chosen by cross-validation (see Section 2.2.3-2.2.5 by Chipman et al. (2010) for detailed information). Furthermore, due to computational reasons, Chipman et al. (2010) recommend to set the number of trees m to 200, which results in excellent predictive performance.

3.5.2 Backfitting MCMC algorithm

Given the aforementioned prior and the observed data, we define the posterior distribution by $p((T_1, M_1), ...(T_m, M_m), \sigma|y)$. Chipman et al. (2010) propose a backfitting Markov Chain Monte Carlo (MCMC) algorithm to sample from this posterior, which relies on a Gibbs sampler. Given the full set of all trees J, we successively draw $(T_j, M_j)|T_{J/j}, M_{J/j}, \sigma, y$ for all $j = 1, ..., m$. Chipman et al. (2010) show that this draw can be conducted in two consecutive steps. Firstly, we draw $T_j|R_j, \sigma$ via the Metropolis-Hastings algorithm, where R_j represents the residuals of the fit excluding the j-th tree. Hereby, we propose a new tree and alter it by rules such as pruning or growing a terminal node (Chipman et al., 2010). Next, we decide whether to accept the new tree based on the ratio of the posterior distributions. The draws of M_j for the terminal nodes consist of independent draws from the normal distribution (Chipman et al., 2010). Secondly, after iterating over all trees, we draw σ from an inverse gamma distribution. Thus, we successively circle between drawing $(T_j$,$M_j)$ conditional on σ and drawing σ conditional on $(T_j$,$M_j)$, providing the Markov Chain (J. Hill & Su, 2013). The Vanilla BART is usually run with a single chain. Additionally, Dorie et al. (2019) propose to combine the results from several chains with distinct starting points, the so-called BART MChains, and we choose ten chains for this method in this research.

Note that BART can also be modified to estimate a probit classification problem denoted by $p(x) \equiv P[Y = 1|x] = \Phi[G(x)]$, where $G(X) \equiv \sum_{j=1}^{m} g(x; T_j, M_j)$ and $\Phi[\cdot]$ is the standard normal cdf. In this setting, we only need to introduce a prior on $(T_j,\ M_j)$ and we follow the same backfitting algorithm as outline above, however including the augmentation idea by Albert and Chib (1993) (see Section 4 by Chipman et al. (2010) for detailed information).

3.5.3 Posterior inference

The successive backfitting algorithm generates a sequence of draws of the sum tree functions $f^*(\cdot) = \sum_{j=1}^{m} g(\cdot; T_j^*, M_j^*)$, which should converge to the true posterior distribution of $f(\cdot)$ (Chipman et al., 2010). To ensure convergence, we adopt the authors' recommendation of 200 burn-ins and 1000 steps per chain. Subsequently, given K samples after burn-in, we can estimate or predict $f(x)$, by computing the posterior mean estimate $f(x) = E[Y|X]$ of the successive sum-of-trees model draws evaluated at any particular x, namely $\frac{1}{K}\sum_{k=1}^{K} f_k^*(x)$. Moreover, the $(1-\alpha)$ posterior confidence intervals are easily obtained, by taking the quantiles of the sample draws. Note that in this set up, we can simply use BART to predict the nuisance parameters $E[Y|D = 1, X]$, $E[Y|D = 0, X]$ and $E[D|X]$, which we can plug into the DML or Generic framework to estimate treatment effects, similar to the Boosting or Random Forest. Alternatively, we can also make use of BART to directly estimate treatment effects.

3.5.4 BART tailored to estimate treatment effects

Given the outcome variable Y_i, treatment variable Z_i and confounding variables X_i, BART can directly estimate treatment effects by fitting the conditional expectation functions $\mathbb{E}[Y_i(1)|X_i = x] = f(1, x)$ and $\mathbb{E}[Y_i(0)|X_i = x] = f(0, x)$. To estimate these functions, we follow the approach of J. L. Hill (2011) and define the sum of trees model by

$$f(z, x) + \epsilon = \sum_{j=1}^{m} g(z, x; T_j, M_j) + \epsilon, \tag{33}$$

where $\epsilon \sim N(0, \sigma^2)$. As outlined above, we impose a prior on each single tree (T_j, M_j) and on the standard deviation σ of the residuals. Next, each step of the successive backfitting MCMC algorithm generates a new draw of $f(\cdot)$ from the posterior distribution (J. Hill & Su, 2013). Thus, at the k-th fixed draw, we retrieve the Individual Treatment Effects denoted by $\tau_i^k = f^k(1, x_i) - f^k(0, x_i)$, for all observations $i = 1, ..., N$.

We calculate the causal effect of interest by taking the average over the corresponding population, for instance the Average Treatment Effect at the k-th draw is given by $\beta_{ate}^k = \frac{1}{N}\sum_{i=1}^{N} \tau_i^k$. Iterating over all $k = 1, ..., K$ draws yields the Monte Carlo approximation to the posterior distribution of the parameter of interest (J. Hill & Su, 2013). Following, we compute the ATE estimate by the posterior mean $\beta_{ate} = \frac{1}{k}\sum_{k=1}^{K} \beta_{ate}^k$. Furthermore, J. L. Hill (2011) propose to form the $(1 - \alpha)$ posterior intervals via normal approximation by the posterior mean plus/minus the critical value times the posterior standard deviation.

We can also change the parameter of interest and retrieve Group Average Treatment Effects. Firstly, for g groups, we calculate the g quantiles of the distribution of the Individual Treatment Effect at the k-th draw, defining observation sets for each group. Then, we compute the GATEs at the k-th draw by $\beta_g^k = \frac{1}{N_g}\sum_{i\in|G|}^{N_g} \tau_i^k$, where $|G|$ denotes such a group set (e.g the most affected group reflected by the top quantile) and N_g reflects the number of observations in this group. Iterating over all draws yields an approximation of the posterior distribution of the GATEs, which allows us to obtain the posterior mean as well as the posterior intervals.

3.5.5 Assessing limited overlap with BART

In contrast to methods relying on the propensity score, BART can address areas with lacking common causal support by using information from the response function. The main concern with lacking common support is whether there is sufficient information to estimate $f(1 - D_i, x_i)$, where D_i denotes the observed treatment variable. If there is not enough information, the posterior standard deviations of the individual-level conditional expectations estimated with BART increase substantially (J. L. Hill, 2011). Thus, we can identify areas with limited overlap by spotting increased standard deviations of the individual specific posterior distribution for each potential outcome, denoted by $o_i^{f_1} = sd(f(1, x_i))$

and $o_i^{f_0} = sd(f(0, x_i))$. Note that in practice, we compute the standard deviations $s_i^{f_1}$ and $s_i^{f_0}$ of the draws of $f(1, x_i)$ and $f(0, x_i)$ within the MCMC-algorithm (J. L. Hill, 2011). Using this information, we define discarding rules, which trim observations evaluated at certain thresholds. The so-called "1-sd-rule" by J. Hill and Su (2013) omits observations above the threshold $s_i^{f_{1-a}} > m_a + sd(s_j^{f_a})$, where $m_a = max_j\{s_j^{f_a}\}, \forall : D_j = a$ and a defines the treatment status. Other alternative options are based on ratios of the posterior standard deviations (see Section 3.1 by J. Hill and Su (2013) for detailed information). However, the authors show that the "1-sd-rule" is most robust across various simulation settings, and thus we implement it within the BART MChains.

4 Data and evaluation criteria

4.1 Data

The data for this research is based on the data analysis challenge “Is Your SATT Where It’s At?”, which was part of the 2016 Atlantic Causal Inference Conference (Dorie et al., 2019). In this competition, 77 different simulations settings, referred to as knobs, are created to neutrally compare the performance of Causal ML methods. While the continuous dependent variable Y_i and the binary treatment assignment D_i are simulated, the covariates X_i are taken from an observational study, to calibrate the data to a real life scenario, namely the Collaborative Perinatal Project (Niswander, 1972).

This project was a longitudinal study on pregnant women and their children, aiming to investigate factors causing development disorders. Specifically, Dorie et al. (2019) consider a hypothetical twin study, examining the impact of birth weight on a child’s IQ and choose covariates which might have been selected by a researcher to capture confounding effects. Given this setup, we obtain 4802 observations and 58 covariates, of which 23 are continuous, three are categorical, five are binary and 27 are non-negative integer. An overview of the covariates is provided in Table 9 in Appendix A.3.

According to Dorie et al. (2019), the data generating process (DGP) of the potential outcomes and the treatment assignment simulation allows to factor the joint distribution conditional on covariates by

$$p(Y(1), Y(0), D|X) = p(Y(1), Y(0)|D, X)p(D|X) = p(Y(1), Y(0)|X)p(D|X), \tag{34}$$

where $p(Y(1), Y(0)|X)$ denotes the response surface and $p(D|X)$ the treatment mechanism assignment. Thus, the unconfoundedness assumption is satisfied, which is essential to identify treatment effects in observational studies, see Section 3.1.

To compare the performance of the methods in various complex scenarios, each simulation setting represents a unique combination of five criteria, specifically (1) the degree of non-linearity in the *response* and *treatment* function, (2) the *percentage of treated*, (3) *overlap*, (4) *alignment* and (5) treatment effect *heterogeneity* (Dorie et al., 2019). Firstly, the degree of non-linearity in the *response* and *treatment* function is specified via a library of generalized additive functions, through which the covariates are transformed and added or multiplied together. Hereby, we have a separate response and treatment composition in each simulation step. As an example, given covariate vector x_i for the i^{th} individual, such a composition of functions with two confounders including interaction and polynomial terms for the response function could be $f(x_i) = f(x_{i1}) + f(x_{i2})^2 + f(x_{i1})f(x_{i2})$, whereas for the treatment function we could have a different composition (also with different confounders), such as $t(x_i) = t(x_{i3})^2 + t(x_{i4})^3$. To test the performance of each method in highly non-linear settings, each function can contain polynomial terms, interactions between covariates and indicator or step functions (Dorie et al., 2019). An exponential link function might also be considered by $g(x_i) = \exp(f(x_i))$. More specifically, linear denotes linear confounding effects, polynomial includes up to third-order polynomial terms, step adds jumps and kinks of the form $I\{x \leq A\}(x_{\cdot j})$ and $(x - B)I\{x \leq C\}(x_{\cdot k})$, and exponential indicates an exponential link functions such as $g(x_i)$ (Dorie et al., 2019). Secondly, the *percentage of treated* indicates the share of observations receiving the treatment and ranges from 35% (low) to 65% (high). Thirdly, full *overlap* indicates that the propensity score is bounded away from zero and one, so that common support is satisfied. To violate this assumption, Dorie et al. (2019) consider simulation settings with penalized overlap, by forcing observations to have a propensity score of zero. Fourthly, *alignment* reflects the degree to which the response and treatment model share the same confounding effects or contain noise. Some confounders might only affect the response model and others only the treatment model, or the same confounder might impact both treatment and response in different functional forms. This complexity can increase the bias in traditional methods, such as linear regressions. To test whether a proposed method can overcome this hurdle, Dorie et al. (2019) specify *alignment* as the marginal probability that a confounder in the treatment assignment mechanism also exists in the response function. Hereby, low indicates an approximate chance of 25% and high a 75% chance. Lastly, *heterogeneity* describes to what extend the treatment effect is interacting with covariates. Hereby, none denotes that the treatment effect is constant conditional on covariates, low denotes interactions with three covariates and high denotes interactions with six covariates (Dorie et al., 2019).

The goal of this research is to challenge and to compare Causal ML methods in scenarios where traditional techniques would fail. Thus, we revisit the 77 knobs, which were the testing ground for the original competition (Dorie et al., 2019). Hereby, a knob defines a unique combination of the aforementioned criteria, for instance knob 1 has a linear *treatment* model, a low *percentage of treated*, penalized *overlap*, a linear *response* model, high *alignment* between the treatment and the response and high *heterogeneity*. For an overview of all 77 knobs we refer to Table 8 in Appendix A.1. Furthermore, for each knob we consider $r = 1, ..., 100$ simulation replications, resulting in a total of 7700 datasets.

4.2 Evaluation criteria

Table 1 provides an overview of the methods used to estimate the target parameters and the criteria considered to evaluate their performance. Firstly, the ATE can be computed by all methods, since we can retrieve the average estimates from all methods which are initially targeting more granular heterogeneous effects. The true ATE is computed for the r-th simulation replication by $\beta_r^{ate} = \frac{1}{N}\sum_{i=1}^{N}(\mathbb{E}[Y_r(1)|X_i] - \mathbb{E}[Y_r(0)|X_i]) = \frac{1}{N}\sum_{i=1}^{N}\tau(X_i)^5$. We evaluate the ATE by the Root Mean Square Error and the coverage rate of interval estimates over all simulation replications R, given by

$$RMSE_{ate} = \sqrt{\sum_{r=1}^{R}(\hat{\beta}_r^{ate} - \beta_r^{ate})^2/R} \qquad cover_{ate} = \frac{1}{R}\sum_{r=1}^{R}\mathbb{1}(\hat{l}_r < \beta_r^{ate} < \hat{u}_r), \tag{35}$$

where l_r and u_r are the lower and upper bounds of the estimated 95% confidence intervals. Additionally, we also calculate the bias by taking the difference between the true parameter and the estimator, as well as the interval length by taking the difference between the upper and the lower bound. We are omitting the ATE analysis for the Double Robust Modified Outcome Method, since it is an intermediate step of the DML. Secondly, we estimate the GATEs with the Generic (GE) Machine Learning for Heterogeneous Treatment Effects, the Causal Forest and BART MChains. The true GATEs per simulation are given by $\tau_r(g) = \int \tau_r(x) f_{X_i|G_i=g}(x)dx$, where g denotes the group membership[5]. Hereby, we choose five groups based on the quantiles of the Individual Treatment Effects. Due to their high practical relevance, we focus on the top 20% quantile, i.e. the most affected group, as well as the bottom 20% quantile, which is the least affected group (Chernozhukov et al., 2018b). Identical to the ATE, we evaluate the GATE using the Root Mean Square Error, coverage rate, bias and confidence interval length. Thirdly, we estimate Individual Treatment Effects using the DR MOM, the Causal Forest and BART MChains. Following Knaus et al. (2018), we compute the ITEs on a validation set by holding out 50% of the data at each replication. The true ITEs are calculated for each replication by $\tau_{r,i}(X_i) = \mathbb{E}[Y_{r,i}(1)|X_i] - \mathbb{E}[Y_{r,i}(0)|X_i]^5$. We evaluate the ITEs by the Precision in the Estimated Heterogeneous Effects (PEHE), or in other words the RMSE of Individual Treatment Effects, for a single replication denoted by $PEHE_r = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\tau}_r(X_i) - \tau_r(X_i))^2}$, where N is the number of individuals (J. L. Hill, 2011). Hereby, we calculate the precision over all replications by taking the mean $\frac{1}{R}\sum_{r=1}^{R} PEHE_r$. Lastly, adopting the evaluation approach by Dorie et al. (2019), the bias, RMSE and interval length are standardized and divided by the standard deviation of the dependent variable Y at each simulation replication.

[5] As proposed by Crump, Hotz, Imbens, and Mitnik (2006), in case we trim observations, we define the subset of the entire covariate space A $\subset$ X excluding trimmed observations and we calculate the causal effects over the subsample i:X$_i \in A$.

Parameter	Evaluation	DML	Generic (GE) ML	Causal Forest	BART MChains	DR MOM
ATE	Bias	x	x	x	x	
	RMSE	x	x	x	x	
	Coverage	x	x	x	x	
	Int length	x	x	x	x	
GATE	Bias		x	x	x	
	RMSE		x	x	x	
	Coverage		x	x	x	
	Int length		x	x	x	
ITE	PEHE			x	x	x

Table 1: Overview of Causal ML methods and evaluation criteria for each treatment effect

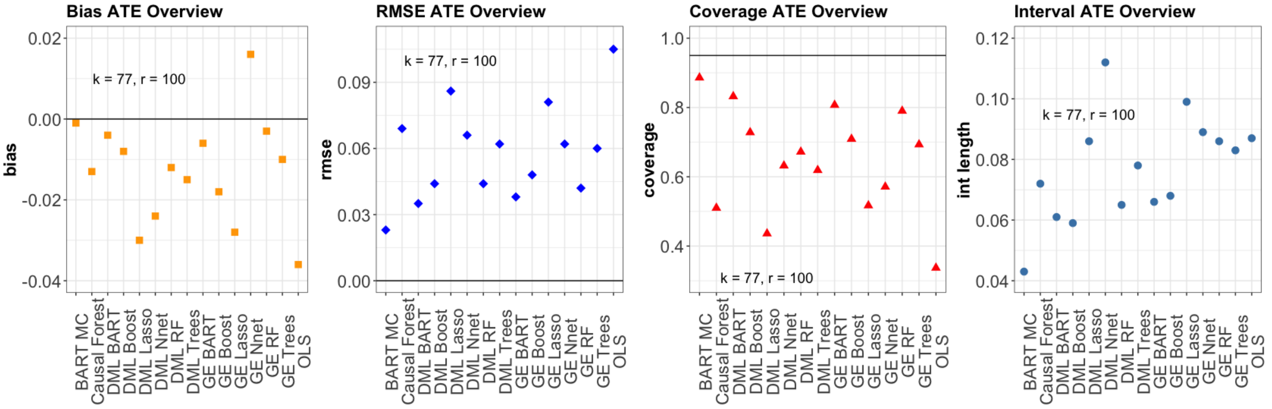
5 Main results

The results section is structured as follows: the first part covers the Average Treatment Effect (ATE), while the second part covers the Heterogeneous Treatment Effects, including Group Average Treatment Effects (GATE) and Individual Treatment Effects (ITE). Within each treatment effect analysis, we first start analyzing the overall performance of each method aggregated over all 77 simulation knobs. Following, we specifically evaluate the impact of each of the five criteria: (1) the functional form of the *treatment* and *response* model, (2) the *percent of treated*, (3) the lack of *overlap*, (4) the level of *alignment* and (5) the level of *heterogeneity*. To disentangle the effects of each criteria separately, we choose a benchmark knob and alter each criteria gradually one at a time. We refer to this as the Benchmarking Analysis. Moreover, we compare the aggregated performances of each criteria at different levels, for instance the aggregated performance of all knobs having low *heterogeneity* versus all knobs having high *heterogeneity*.

5.1 Average Treatment Effect

Figure 1 and the corresponding Table 2 show an overview of the performance of all methods, estimating the ATE over 77 simulation settings with 100 replications each. In Figure 1a, squares reflect bias and diamonds Root Mean Square Errors, while in Figure 1b, triangles reflect coverage and circles interval lengths. All evaluation criteria are averaged across all simulation settings.

The overall best performing method is BART MChains with the lowest bias (-0.001), the lowest Root Mean Square Error (0.02), the highest coverage (0.89) and a relatively small interval length (0.04). The second best performing group is the new proposed DML BART and GE BART, as they have a higher coverage than all other ML techniques within the DML and Generic framework (0.83 and 0.81) and show competitive performance when considering all other evaluation criteria. The third best performing group contains DML Boosting, DML Random Forest, GE Boosting and GE Random Forest. The last group contains Causal Forest, DML Lasso, DML Trees, DML Nnet, GE Lasso, GE Trees and GE Nnet. All of these methods have a moderate bias and RMSE, however the coverage rate of the Causal Forest is exceptionally low (0.51). The Lasso-based methods are the worst performing ones, with high RMSEs (DML Lasso: 0.09 and GE Lasso: 0.08) and low coverage rates (DML Lasso: 0.44 and GE Lasso: 0.52). Lastly, the OLS is clearly outperformed by all methods, as it has the greatest bias and Root Mean Square Error, as well as the lowest coverage. This was to be expected, since the OLS estimator cannot conform with the complex simulation criteria, such as polynomial, step or exponential *response/treatment* functions, high *alignment* and high *heterogeneity*. The impact of each criteria on the performance of OLS follows in the Benchmarking Analysis.



(a) Bias/RMSE

(b) Coverage/Interval Length

Figure 1: Overview of estimation of ATE, averaged across all 77 simulation knobs

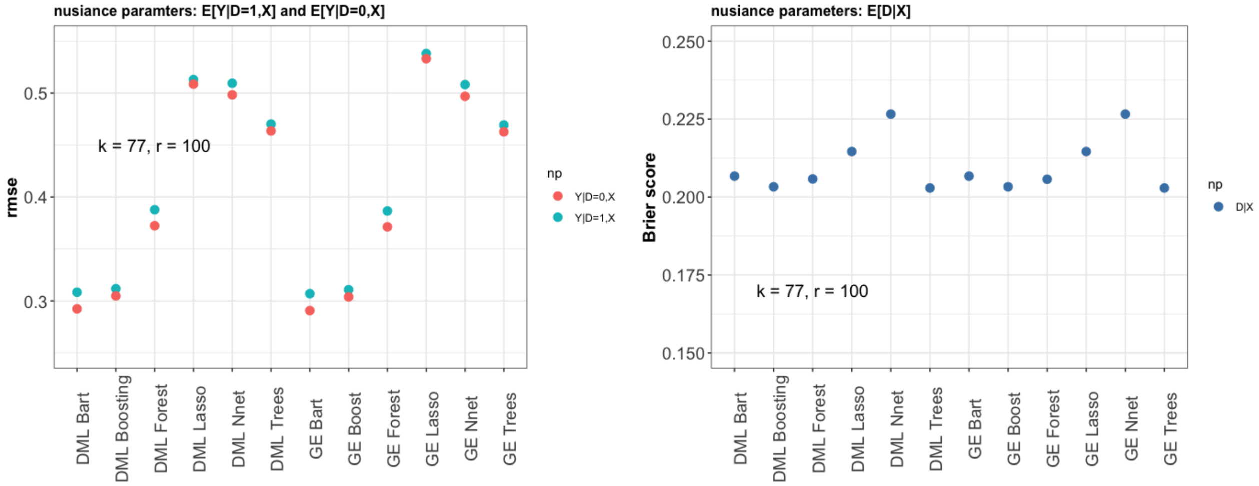
Note: This figure shows the performance of all methods in estimating the ATE, with results averaged across all 77 simulation knobs, with 100 replications per knob. Thus, in total we consider 7700 datasets. In Panel A, squares reflect bias, diamonds Root Mean Square Errors, while in Panel B, triangles reflect coverage and circles interval lengths. The horizontal lines reflect the desired theoretical values of the bias, RMSE and the coverage rate.

crit- eria	BART MC	Causal Forest	DML BART	DML Boost	DML Lasso	DML Nnet	DML RF	DML Trees	GE BART	GE Boost	GE Lasso	GE Nnet	GE RF	GE Trees	OLS
bias	-0.001	-0.01	-0.004	-0.01	-0.03	-0.02	-0.01	-0.01	-0.01	-0.02	-0.03	0.02	-0.003	-0.01	-0.04
rmse	0.02	0.07	0.04	0.04	0.09	0.07	0.04	0.06	0.04	0.05	0.08	0.06	0.04	0.06	0.10
cov	0.89	0.51	0.83	0.73	0.44	0.63	0.67	0.62	0.81	0.71	0.52	0.57	0.79	0.69	0.34
int	0.04	0.07	0.06	0.06	0.09	0.11	0.07	0.08	0.07	0.07	0.10	0.09	0.09	0.08	0.09

Table 2: Overview of estimation of ATE, averaged across all 77 simulation knobs

Chipman et al. (2010) argue that BART is less sensitive to the choice of tuning parameters compared to the other statistical learning techniques, which could in part explain the overall superior performance of BART-based methods. Interestingly, BART MChains seems to perform slightly better than BART used within the DML and GE framework. A possible explanation could be that the DML and Generic framework apply sample splitting to ensure valid theoretical results. Thus, even though we repeat over several sample splits and we use cross-fitting to keep full efficiency, we increase uncertainty as we only use a subset of the data for estimation. In contrast, BART MChains estimates on the full sample. Another reason could be that BART MChains combines several Markov Chains, whereas the Vanilla BART used within the DML and Generic framework only uses a single chain. Using several chains increases the sample size of the posterior distribution which, as shown in the results of Dorie et al. (2019), improves the performance compared to using a single chain.

To understand why the Machine Learning methods within the DML and the Generic framework perform differently, we analyze their performance in predicting the nuisance parameters. Figure 2a shows the RMSEs of the response functions ($E[Y|D=1,X]$ and $E[Y|D=0,X]$) and Figure 2b shows the Brier score[6] of the propensity function ($E[D|X]$). We observe substantial differences in the ML methods used within the DML and Generic framework when predicting the conditional expectations of the response function. In particular, BART performs best, closely followed by Boosting, indicating that growing a sequence of trees using a weak learner approach, as outlined in Section 3.5 and Appendix A.6.3, seems to be most suitable. The third best performing method is DML/Generic Forest, which also has the advantage of growing a multitude of trees on subsets of the data to avoid overfitting and to lower the variance compared to single trees. However, the random splitting mechanism does not seem to be as strong in modeling the response function as the aforementioned weak learner approach. In general, the Tree-based structure appears to be most appropriate, since even the single tree versions perform better than Lasso and Neural Net. Schiltz, Masci, Agasisti, and Horn (2018) highlight that trees have the ability to pick up multiple discontinuous or even non-linear interaction terms by construction. Figure 2b shows that the ML methods within the DML and Generic framework result in a similar pattern when predicting the propensity score. The Tree-based methods (BART, Boosting, Random Forest, Trees) perform best and mostly in-line, while Lasso and Neural Net perform slightly worse. However, we observe that the differences between the methods are much less pronounced.



(a) RMSE response functions

(b) Brier score propensity score

Figure 2: Performance nuisance parameters

Note: This figure shows the performance of all methods in predicting the nuisance parameters. Panel A presents the RMSEs of the response functions ($E[Y|D=1,X]$ and $E[Y|D=0,X]$), while Panel B shows the Brier score of the propensity function ($E[D|X]$). In this figure, k refers to the number of knobs and r the number of simulation replications.

[6] We consider the Brier score by Brier (1950), because it is suited to evaluate probabilistic predictions (p_i) with categorical outcomes (o_i) denoted by $BS = \frac{1}{N}\sum_{i=1}^{N}(p_i - o_i)^2$, which we substitute with the predicted propensity score and the observed binary treatment variable. Effectively, the Brier score is a MSE between the probability and the categorical outcome, where a value of zero is the best score achievable.

In contrast to the traditional Tree structure, the Causal Forest is rather tailored to capture Heterogeneous Treatment Effects, since its splitting criterion in Equation (22) is maximizing heterogeneity, which could explain why the method is not among the best performing ones in estimating the ATE. Furthermore, the simulation studies of Wager and Athey (2018) reveal that the coverage rates of the Causal Forest are decreasing with an increasing amount of confounders, which questions whether the asymptotic theory is valid given the dimension of the data used in this research ($n = 4802$, $p = 58$).

Overall, the top performing methods (BART MChains, DML BART, DML Boosting, DML Random Forest, GE BART, GE Boosting and GE Random Forest) perform reasonably well considering the bias and RMSE, since the absolute values of both metrics are at or below 0.05, which is relatively low compared to traditional estimations techniques (Cohen, 1962; Dorie et al., 2019). However, none of the methods reaches the intended nominal coverage of 95% averaged across all knobs. A similar pattern of results is obtained in the competition by Dorie et al. (2019), presented in Figures 15 and 16 in Appendix A.2. Even though the competition focuses on estimating the ATT rather than ATE and hence does not qualify for direct comparison of the results, it does allow us to confirm the main observations stemming from our results. Namely, the competition results show that the absolute values of the bias and RMSE of the better performing methods are also at or below 0.05 and none of the methods reaches nominal coverage. Moreover, BART MChains is in the top preforming methods, reaching similar values when estimating the ATT and the ATE.

To analyze the impact of each criteria separately, we choose knob 27 as a benchmark, having a polynomial *treatment* model, low *percentage of treated*, full *overlap*, step *response* model, low *alignment* and low *heterogeneity*[7]. Subsequently, we vary each criteria one at a time, as displayed in Table 3.

knob	treatment model	percent treated	overlap	response model	alignment	heterogeneity
27	*polynomial*	*low*	*full*	*step*	*low*	*low*
55	**step**	low	full	step	low	low
40	polynomial	**high**	full	step	low	low
21	polynomial	low	**penalize**	step	low	low
31	polynomial	low	full	**exponential**	low	low
29	polynomial	low	full	step	**high**	low
28	polynomial	low	full	step	low	**high**

Table 3: Benchmarking Analysis: altering each criteria gradually

Table 4 shows the results of the RMSE and coverage, while Table 10 in Appendix A.4.1 provides the results of the bias and the interval length. Firstly, we note that in the benchmark knob all methods apart from the OLS have an RMSE below 0.05 and considerably higher coverage rates than in the aggregated overview across all knobs in Figure 1. Even more remarkably, BART MChains, DML BART, GE BART and GE Boosting reach the nominal coverage of 0.95, while DML Boosting comes

[7] It might be more intuitive to choose a knob with the simplest combination as benchmark, for instance by including a linear *response*/*treatment* function and no *heterogeneity*. However, such a knob is not available, given that the competition includes only highly complex settings where traditional methods would fail. As a result, we choose knob 27 as benchmark, considering that it is one of the simplest knobs available due to the full *overlap* and the low *heterogeneity* level.

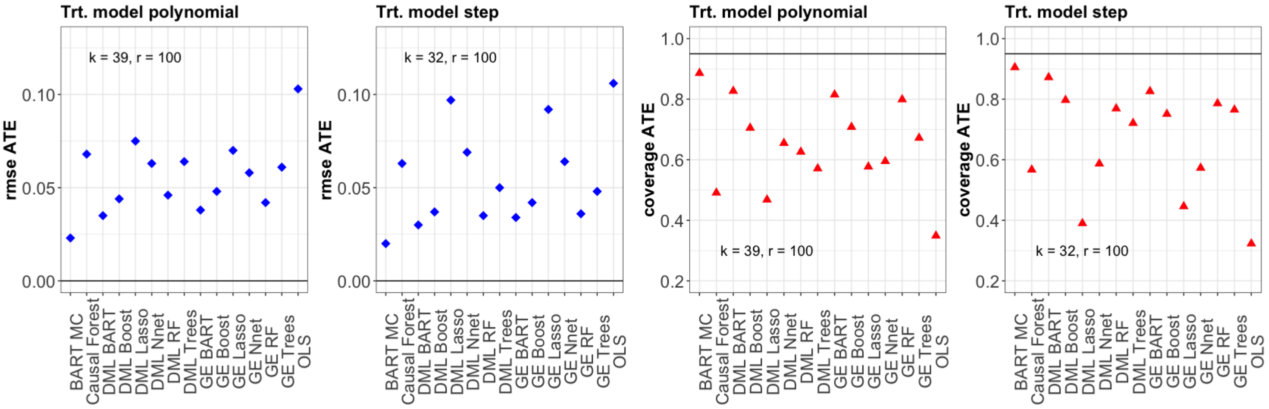
close with a coverage of 0.93. Hereby, BART MChains has a relatively small interval length of 0.03, while methods of the DML and the Generic framework have a moderate interval length of 0.05 (see Appendix A.4.1 Table 10).

knob	BART MC	Causal Forest	DML BART	DML Boost	DML Lasso	DML Nnet	DML RF	DML Trees	GE BART	GE Boost	GE Lasso	GE Nnet	GE RF	GE Trees	OLS
27	0.01 (0.97)	0.02 (0.87)	0.01 (0.96)	0.01 (0.93)	0.04 (0.74)	0.03 (0.85)	0.02 (0.89)	0.02 (0.86)	0.01 (0.96)	0.01 (0.96)	0.04 (0.84)	0.04 (0.70)	0.02 (0.83)	0.02 (0.88)	0.07 (0.67)
55	0.01 (0.97)	0.03 (0.79)	0.01 (0.98)	0.01 (0.95)	0.06 (0.72)	0.04 (0.86)	0.01 (0.97)	0.02 (0.93)	0.01 (0.98)	0.01 (0.97)	0.06 (0.78)	0.05 (0.60)	0.02 (0.73)	0.02 (0.97)	0.08 (0.69)
40	0.01 (0.98)	0.03 (0.80)	0.01 (0.96)	0.01 (0.92)	0.04 (0.73)	0.03 (0.84)	0.02 (0.87)	0.02 (0.87)	0.01 (0.95)	0.01 (0.95)	0.04 (0.84)	0.04 (0.72)	0.02 (0.83)	0.02 (0.92)	0.06 (0.63)
21	0.01 (0.92)	0.03 (0.68)	0.03 (0.78)	0.03 (0.73)	0.04 (0.70)	0.04 (0.86)	0.03 (0.71)	0.03 (0.77)	0.03 (0.85)	0.04 (0.72)	0.04 (0.80)	0.04 (0.68)	0.03 (0.84)	0.03 (0.84)	0.05 (0.64)
31	0.01 (0.92)	0.03 (0.83)	0.01 (0.99)	0.01 (0.96)	0.04 (0.74)	0.03 (0.92)	0.02 (0.95)	0.02 (0.88)	0.01 (0.92)	0.01 (0.96)	0.04 (0.78)	0.04 (0.61)	0.02 (0.82)	0.02 (0.96)	0.06 (0.65)
29	0.01 (0.98)	0.05 (0.49)	0.01 (0.90)	0.02 (0.68)	0.06 (0.42)	0.06 (0.53)	0.03 (0.63)	0.04 (0.46)	0.01 (0.94)	0.01 (0.87)	0.05 (0.52)	0.04 (0.64)	0.02 (0.87)	0.04 (0.68)	0.11 (0.31)
28	0.01 (0.92)	0.04 (0.59)	0.02 (0.89)	0.02 (0.83)	0.07 (0.38)	0.06 (0.58)	0.03 (0.65)	0.04 (0.69)	0.01 (0.91)	0.01 (0.92)	0.07 (0.57)	0.05 (0.52)	0.02 (0.84)	0.03 (0.81)	0.12 (0.16)

Table 4: Benchmarking Analysis results ATE: RMSE and (Coverage)

Notes: This table summarizes the RMSEs and the corresponding coverage rates in parenthesis of all methods when estimating the ATE, with 100 replications per simulation knob. We refer to this as the Benchmarking Analysis, whereby knob 27 is taken as benchmark and all other knobs vary each criteria one at a time, as shown in Table 3.

When altering the *treatment* model from polynomial to step (knob 55), the Lasso-based methods perform worse, with increasing RMSEs and decreasing coverage rates. In contrast, the coverage rate of the DML Random Forest increases from 0.89 to 0.97 and DML Trees from 0.86 to 0.93. The overview in Figure 3 also confirms that Lasso-based methods perform much worse when facing step functions, whereas the Tree-based methods show improvements, especially with respect to coverage. This result makes sense intuitively, since Lasso is able to capture polynomial terms, however its functional form cannot incorporate step functions. In contrast, a Tree contains splitting rules based on certain cutoff points that divide the covariate space into distinct sets. Specifically, we split the covariate space into the regions $\{X|X_j < s\}$ and $\{X|X_j > s\}$ at each iteration of the Tree-algorithm, where X_j is the predictor whose cutoff point s minimizes the error rate the most (Friedman et al., 2001). Since we predict different values in these sub-regions, we can model discontinuous functional forms, such as step functions. Altering the *response* model from step to exponential (knob 31) does not reveal any notable changes. Lastly, across all knobs with polynomial, step or exponential *treatment/response* functions, OLS is consistently outperformed by the Causal ML functions, which is to be expected as the linearity assumption is violated. Since non-linearities are a common problem in empirical applications, we generally recommend to use the new proposed Causal ML methods due to their higher flexibility. Even more so, practitioners can use them to detect hidden complexity in the functional forms in the first place, by directly comparing their estimates with those of the traditional approaches and evaluate whether the results are in-line.

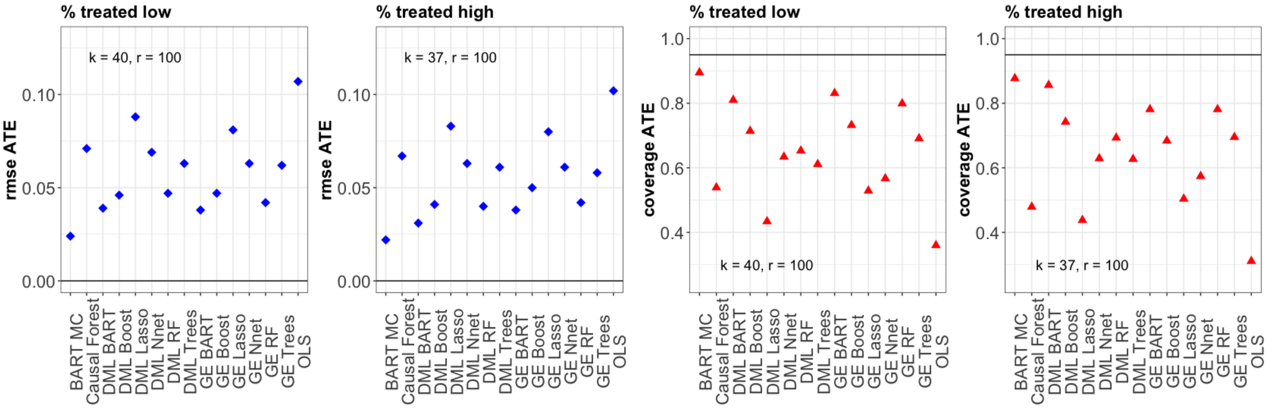


(a) RMSE (b) Coverage

Figure 3: Trt. model polynomial/step

Note: This figure shows the RMSEs and coverage rates of the ATE estimation, where results are averaged across all knobs with a polynomial *treatment* model versus all knobs with a step *treatment* model. r denotes the number of replications per simulation knob and k is the number of knobs with a given *treatment* model type. As the sum of both ($k = 39$ and $k = 32$) is below the total number of knobs (77), there are remaining knobs with a different *treatment* model, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the RMSE and the coverage rate.

Changing the *percent of treated* from low to high (knob 40) does not change the performance of most of the Causal ML methods with respect to the benchmark, only the RMSE of the Causal Forest increases from 0.02 to 0.03 and its coverage decreases from 0.87 to 0.80. When comparing all simulations with low and high *percent of treated* in Figure 4, we also observe similar results. This observation is reassuring for practitioners, given that in observational studies it can be costly or infeasible to obtain a fair 50-50 split.



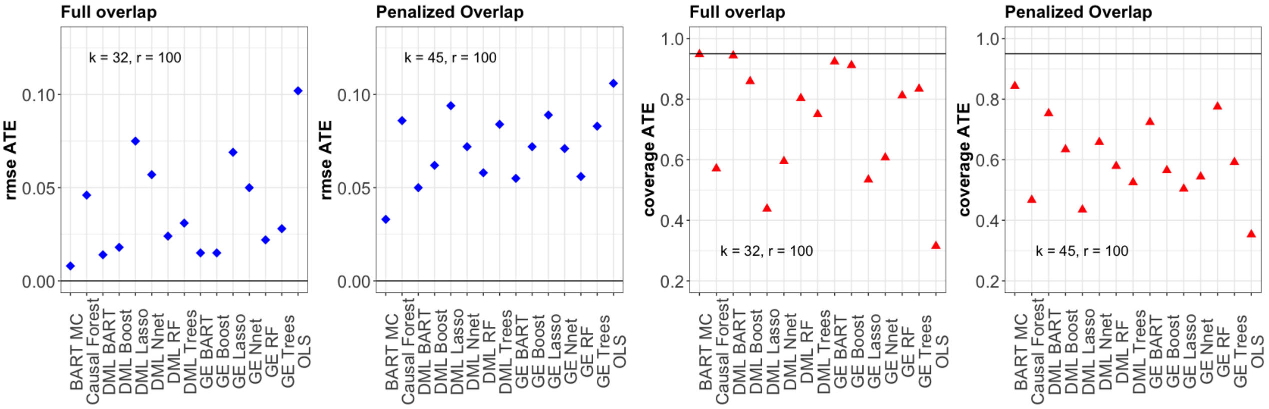
(a) RMSE (b) Coverage

Figure 4: Low/High % treated

Note: This figure shows the RMSEs and coverage rates of the ATE estimation, where results are averaged across all knobs with low *% treated* versus all knobs with a high *% treated*. r denotes the number of replications per simulation knob and k is the number of knobs with a given *% treated*. The horizontal lines reflect the desired theoretical values of the RMSE and the coverage rate.

Violating the common support assumption by moving from full to penalized *overlap* (knob 21) increases the RMSEs of all Causal ML methods and almost all methods show a considerable drop in coverage. That becomes particularly evident for the DML/GE BART- and Boosting-based methods, as the coverage of DML BART decreases from 0.96 to 0.78, GE BART from 0.96 to 0.85, DML Boosting from 0.93 to 0.73 and GE Boosting from 0.96 to 0.72. The overview across all knobs in Figure 5 confirms the same observation. Remarkably, Figure 5 shows that when there is full *overlap*, next to BART MChains, the new proposed DML BART, which combines the predictive strength of BART with the theoretical results of the DML framework, reaches nominal coverage. When the overlap assumption is violated, although loosing nominal coverage, BART MChains is the best performing method, highlighting that its performance is less severely affected compared to the other methods. Figure 5 also confirms that the best performing methods of the DML and Generic framework appear to be more negatively affected than BART MChains by the lack of *overlap.* Lastly, although at first glance DML Neural Net and GE Random Forest seem robust in terms of the coverage rates, Figure 20 in Appendix A.4.1 shows that this comes at the the cost of extremely wide confidence intervals.

Several researchers have shown that the violation of the overlap assumption leads to imprecise Average Treatment Effects, as well as distorted confidence intervals (Crump et al., 2006; Rothe, 2017). Clearly, we observe a similar pattern with the proposed Causal ML methods. For that reason, we investigate how each method is addressing the common support. In the DML and the Generic framework, we disproportionate the impact of extreme propensity score weights, by trimming the predicted propensity scores below 0.01 or above 0.99 at each split (Chernozhukov et al., 2018a; Chernozhukov, Demirer, et al., 2017). This technique bears the risk that we trim the incorrect observations (or even not trim at all), if the propensity score is not estimated correctly. Causal Forest also deals with the lack of *overlap*, by introducing weights based on the propensity score (Li et al., 2018). However, given that the performance of the Causal Forest in estimating the ATE is not strong in the first place, little can be assessed on its functionality to address *overlap.* As outlined in Section 3.5.5, instead of relying on the propensity score, BART can identify areas with penalized *overlap* based on the response function, since the standard deviations of the posterior distribution increase in areas of lacking common support, and based on this information we omit observations using the "1-sd-discard" rule (J. Hill & Su, 2013). This approach bears the risk that we trim incorrect observations, if the response function is not estimated correctly. However, given the results, the response function seems to be superior to using the propensity score when accounting for lacking *overlap* in the simulation settings of this research. Given that limited *overlap* is an often reoccurring issue in the estimation of the ATE under unconfoundedness in applied research (Crump et al., 2006), we recommend to use methods that identify the problematic covariate space by using information from the response function and to trim observations based on data-driven thresholds. Methods relying on the propensity score could also be a competitive alternative, however the practitioner should ensure that its estimation is correct and that hard-coded threshold are chosen carefully.



(a) RMSE

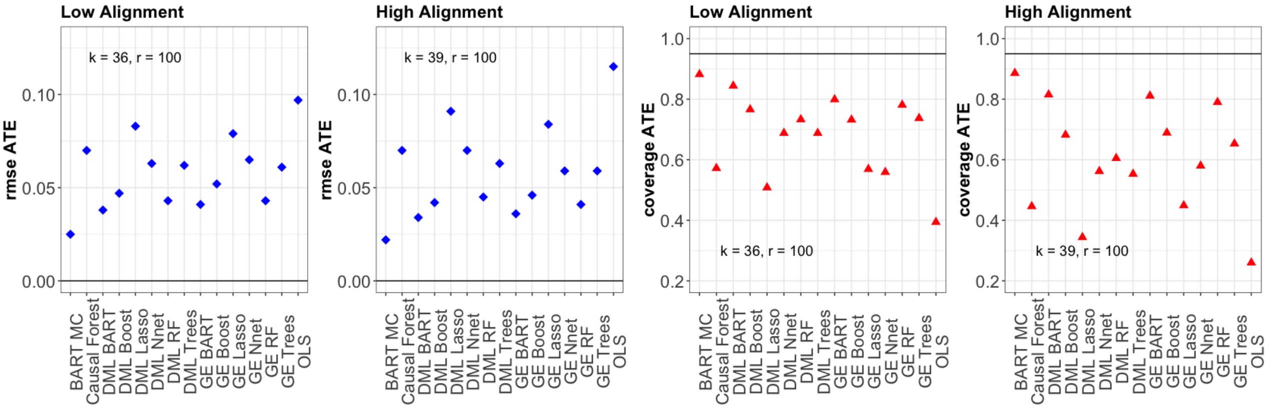
(b) Coverage

Figure 5: Full/Penalized overlap

Note: This figure shows the RMSEs and coverage rates of the ATE estimation, where results are averaged across all knobs with full *overlap* versus all knobs with penalized *overlap*. r denotes the number of replications per simulation knob and k is the number of knobs with a given *overlap*. The horizontal lines reflect the desired theoretical values of the RMSE and the coverage rate.

When increasing the level of *alignment* from low to high (knob 29), the RMSEs of the Causal ML methods remain rather stable. However, we see a mixed behavior with respect to the coverage rates. While the coverage rates of the BART MChains, DML BART, GE BART and GE Forest are relatively robust, those of the Causal Forest drop substantially (from 0.87 to 0.49) and some of the other methods decrease as well (DML Lasso from 0.74 to 0.42 or DML Random Forest from 0.89 to 0.63). Furthermore, the OLS estimator also becomes considerably worse, since its RMSE increases from 0.07 to 0.11 and its coverage rate drops from 0.67 to 0.31. The overview across all knobs in Figure 6 confirms these observations. Additionally, for the methods with decreased coverage, Figure 21 in Appendix A.4.1 shows an increase in bias, which is likely to cause a shift of the confidence intervals.

Dorie et al. (2019) state that the covariates that are relevant both in the assignment and the response function (which is the case with high *alignment*) have the potential to cause bias in the estimation of treatment effects, since they might enter both equations with different functional forms. For instance, a confounding effect might play a role in the response function as a linear term and in the treatment function as square term. Thus, the OLS performs worse in complex alignment settings, since the response, treatment and all control variables are modelled in one regression equation, without variable selection and separate estimation of the propensity score is missing. In contrast, the Causal ML methods should be more robust, since they incorporate an estimation step of both the response and treatment function, including variable selection. Thus, it is surprising that some of the Causal ML methods are negatively affected in terms of bias and coverage rates. Hence, we acknowledge that this topic requires future research. A possible starting point could be to investigate whether the variable selection or weighting mechanism within each ML method are identifying the right confounders (for instance the covariates chosen by Lasso or the covariates contributing to the upper splits of decision trees).



(a) RMSE

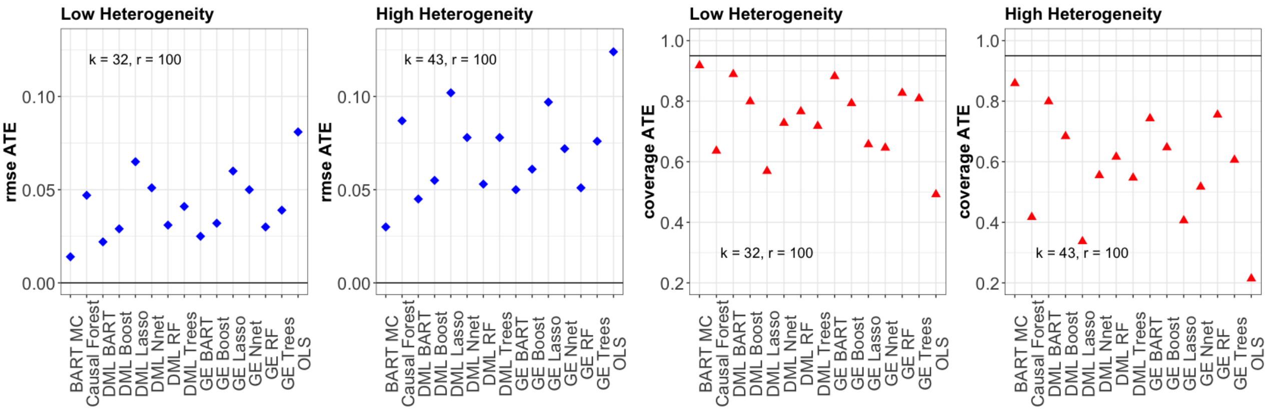
(b) Coverage

Figure 6: Low/High alignment

Note: This figure shows the RMSEs and coverage rates of the ATE estimation, where results are averaged across all knobs with low *alignment* versus all knobs with high *alignment*. r denotes the number of replications per simulation knob and k is the number of knobs with a given level of *alignment*. As the sum of both ($k = 36$ and $k = 39$) is below the total number of knobs (77), there are remaining knobs with a different *alignment* type, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the RMSE and the coverage rate.

When increasing the level of *heterogeneity* from low to high (knob 28), most of the methods preform worse in estimating ATE. This result makes intuitive sense, since we increase the complexity and non-linearity of the model. However, we can still distinguish between marginally impacted methods and methods that perform significantly worse. On the one hand, ensemble methods which grow a multitude of trees seem more robust. BART MChains, GE BART, GE Boosting and GE Random Forest remain stable in terms of RMSE and maintain high coverage rates. DML Boosting, DML Random Forest and Causal Forest only increase slightly in terms of RMSE, but the coverage rates are decreasing to a larger extent (0.93 to 0.83, 0.89 to 0.65, 0.87 to 0.59, respectively). On the other hand, Lasso- and Neural Net-based methods do not seem to address the heterogeneity properly. DML Lasso reaches the maximum RMSE (0.07) and the minimum coverage (0.38) among all Causal ML methods within the Benchmarking Analysis. The OLS is also severely affected by the increased heterogeneity, since the RMSE increases from 0.07 to 0.12 and the coverage drops from 0.67 to 0.16. The overview of all knobs with low and high *heterogeneity* in Figure 7 shows similar patterns as the Benchmarking Analysis. Tree-based ensemble Causal ML methods seem to be most robust, while the OLS becomes substantially worse.

The superior performance of the Tree-based methods demonstrates their ability to pick up complex interactions by construction (Schiltz et al., 2018). In contrast, the reason for the worse performance of the OLS lies within its implementation. Namely, we only include the control variables X_i and do not add any interactions terms with the treatment ($D_i \times X_i$), see Equation 1. In applied econometric research, it is common practice to capture heterogeneity by adding these interaction terms between the treatment (D_i) and the control variables (X_i) (Hainmueller, Mummolo, & Xu, 2019). We acknowledge that this might improve the estimate, if the researcher knows ex-ante which control variables are interacting with the treatment. However, if this specific knowledge is missing, the only way to add the correct interactions terms is by trying out various regression combinations. Such an approach is infeasible in this research, as the covariates causing the heterogeneity alter at each simulation replication. Adding all interaction terms at the same time is also not viable, since it increases the dimensions substantially and leads to imprecise results (Green & Kern, 2012). Furthermore, the linearity assumption of OLS implies that the heterogeneous effects have a linear relationship, which is violated in case they are based non-linear functional forms (Hainmueller et al., 2019). Given these results, we recommend to use Tree-based ensemble Causal ML methods that are able to pick up complex heterogeneous effects by construction, even if the target parameter is the ATE.



(a) RMSE

(b) Coverage

Figure 7: Low/High heterogeneity

Note: This figure shows the RMSEs and coverage rates of the ATE estimation, where results are averaged across all knobs with low *heterogeneity* versus all knobs with high *heterogeneity*. r denotes the number of replications per simulation knob and k is the number of knobs with a given level of *heterogeneity*. As the sum of both ($k = 32$ and $k = 43$) is below the total number of knobs (77), there are remaining knobs with a different *heterogeneity* type, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the RMSE and the coverage rate.

The previously discussed results use the tuning parameters shown in Appendix A.6.6 Table 17. However, Green and Kern (2012) point out that unlike BART, the performances of the Machine Learning techniques such as Boosting, Random Forest or Neural Net are much more sensitive to the choice of these tuning parameters. Thus, to evaluate whether the results are robust given the set parameters, we conduct a sensitivity check and replicate the Benchmarking Analysis, whereby we chose the input parameters of Lasso, Tree, Boosting, Random Forest and Neural Net via a grid search, see Appendix Section A.7 for more details. When comparing the results of the ATE Benchmarking Analysis (Table 4 and Appendix A.4.1 Table 10) with the results of the sensitivity check (Appendix A.7 Table 23 and Appendix A.7 Table 24), we observe that the coverage rates of the ML methods improve slightly when the tuning parameters are chosen by a grid search, however the results are overall stable, confirming that the choice of tuning parameters in the Appendix A.6.6 Table 17 is appropriate[8].

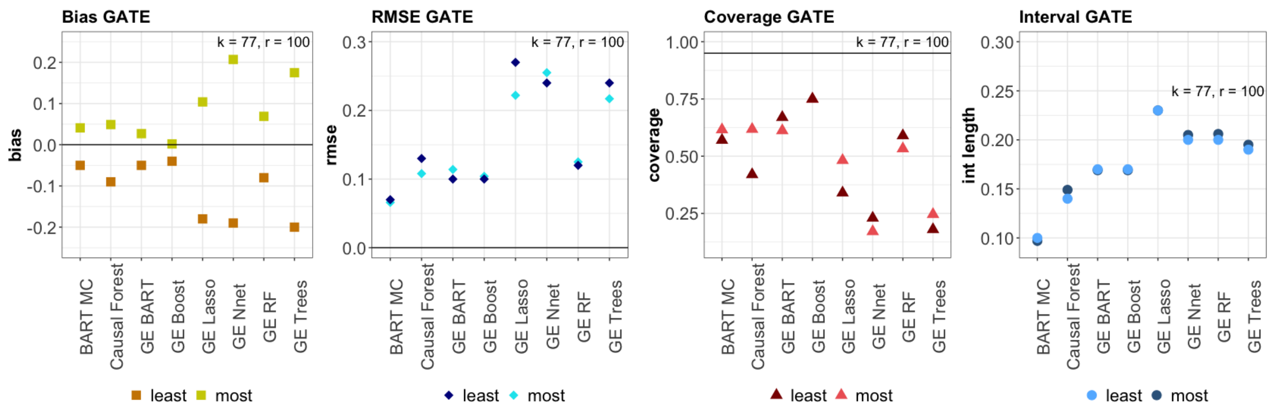
[8] Since tuning the ML input parameters over a grid search is computationally much more intensive, we implement the sensitivity analysis only for the knobs of the ATE Benchmarking Analysis and not for the all 77 knobs of the competition.

5.2 Heterogeneous Treatment Effects

In the following, we analyze the performance of the methods in estimating the Heterogeneous Treatment Effects. Firstly, we analyze the Group Average Treatment Effects, whereby we focus on the estimation of the 20% most and 20% least affected quantiles. These groups are of high interest for practitioners, as they disclose the effect of the treatment on the most and least affected groups. (Chernozhukov et al., 2018b). Secondly, we consider the performance of all methods when estimating Individual Treatment Effects.

5.2.1 Group Average Treatment Effects

Figure 8 and Table 5 provide an overview of the bias, RMSE, coverage and interval length of the most and the least affected group effects, averaged across all 77 simulation knobs. Overall, we observe a higher bias and RMSE, lower coverage and wider intervals for all methods compared to the estimation of ATE (see Figure 1), demonstrating that capturing Heterogeneous Treatment Effects is a more challenging task. When analyzing the bias and RMSE, we can distinguish between two groups. On the one hand, BART MChains, Causal Forest, GE BART, GE Boosting and GE Random Forest are clearly the best performing methods. Within this group, the ensemble methods, which grow a sequence of trees based on a weak learner approach, perform best. BART MChains shows the lowest RMSE (most: 0.07, least: 0.07) and shortest interval length (most: 0.1, least: 0.1), while GE Boosting shows the lowest bias (most: 0.00, least: -0.04) and the highest coverage (most: 0.75, least: 0.75). On the other hand, GE Lasso, GE Nnet and GE Trees perform significantly worse. The different performance of the ML methods within the Generic framework can also be explained in part by the predictive strength in estimating the nuisance parameters, see Figure 2. The advantage of growing multiple trees becomes more evident when estimating Heterogeneous Treatment Effects, since the Generic Tree-based ensemble methods considerably outperform Generic (single) Tree in the GATE, compared to the ATE analysis. Furthermore, considering all four criteria, the Causal Forest ranks among the top performing methods. That was not the case in the ATE analysis, which confirms that the method is better suited to estimate Heterogeneous Treatment Effects, in-line with the splitting criterion within the framework to maximize heterogeneity (see Equation (22)). Interestingly, we also observe that the Causal Forest clearly results in lower RMSE and higher coverage for the most affected group, compared to the least affected one. In contrast, methods such as BART MChains and GE Boosting seem to be more balanced and with nearly identical performance with respect to both groups.



(a) RMSE (b) Coverage

Figure 8: Overview of estimation of GATEs, averaged across all 77 simulation knobs

Note: This figure shows the performance of all methods in estimating the GATEs, with results averaged across all 77 simulation knobs (k), with 100 replications (r) per knob. Thus, in total we consider 7700 datasets. In Panel A, squares reflect bias, diamonds Root Mean Square Errors, while in Panel B, triangles reflect coverage and circles interval lengths.

criteria	BART MC		Causal Forest		GE BART		GE Boost		GE Lasso		GE Nnet		GE RF		GE Trees	
	most	least	most	least	most	least	most	least	most	least	most	least	most	least	most	least
bias	0.04	-0.05	0.05	-0.09	0.03	-0.05	0.00	-0.04	0.10	-0.18	0.21	-0.19	0.07	-0.08	0.18	-0.20
rmse	0.07	0.07	0.11	0.13	0.11	0.10	0.10	0.10	0.22	0.27	0.26	0.24	0.12	0.12	0.22	0.24
cov	0.62	0.57	0.62	0.42	0.61	0.67	0.75	0.75	0.48	0.34	0.17	0.23	0.53	0.59	0.25	0.18
int	0.10	0.10	0.15	0.14	0.17	0.17	0.17	0.17	0.23	0.23	0.21	0.20	0.21	0.20	0.19	0.19

Table 5: Overview of estimation of GATEs, averaged across all 77 simulation knobs

To analyze the specific effect of each criteria on the performance of the methods in estimating the most and the least affected groups, we conduct a Benchmarking Analysis for the GATEs. We choose the same benchmark knob as in the ATE analysis, however we change the level of *heterogeneity* in the benchmark to high (see Table 6), as the primary focus now is to test how well the methods are capturing heterogeneity.

knob	treatment model	percent treated	overlap	response model	alignment	heterogeneity
28	*polynomial*	*low*	*full*	*step*	*low*	*high*
56	**step**	low	full	step	low	high
41	polynomial	**high**	full	step	low	high
22	polynomial	low	**penalize**	step	low	high
32	polynomial	low	full	**exponential**	low	high
30	polynomial	low	full	step	**high**	high
27	polynomial	low	full	step	low	**low**

Table 6: Benchmarking GATE: altering each criteria gradually

Table 7 shows the RMSE and the coverage, while Table 11 in Appendix A.4.2 shows the corresponding bias and interval length. In the benchmark setting (knob 28), BART MChains shows the lowest RMSE (most: 0.05, least: 0.06) and GE Boosting has the highest coverage (most: 0.79, least: 0.78). Causal Forest, GE BART and GE Random Forest also perform reasonably well in terms of RMSE, however the coverage rates are lower. In-line with the overview analysis, the performance of GE Lasso, GE

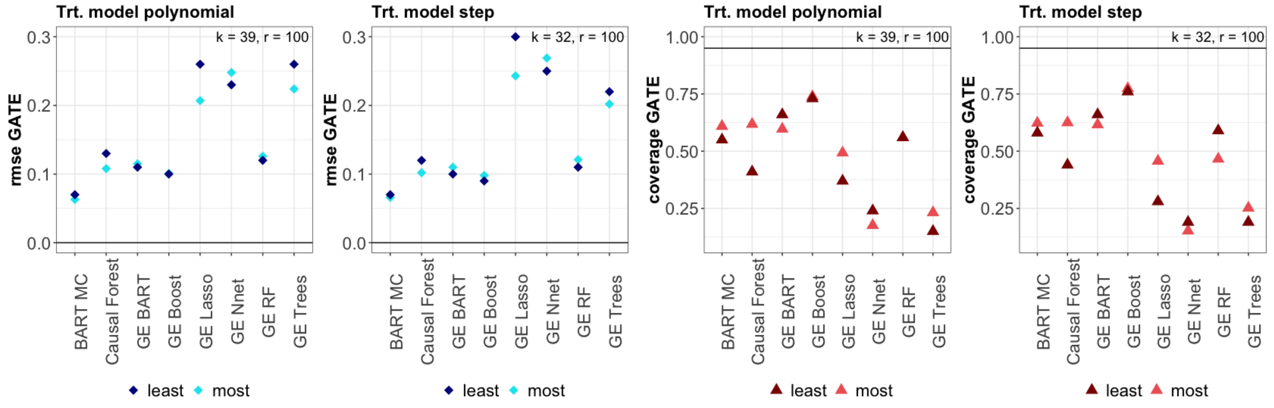
Neural Net and GE Trees is clearly deteriorating and the methods do not seem suitable for estimating the effect on the most and least affected group.

knob	BART MC		Causal Forest		GE BART		GE Boost		GE Lasso		GE Nnet		GE RF		GE Trees	
	most	least	most	least	most	least	most	least	most	least	most	least	most	least	most	least
28	0.05	0.06	0.10	0.10	0.10	0.08	0.07	0.07	0.23	0.26	0.27	0.22	0.11	0.10	0.19	0.21
	(0.66)	(0.64)	(0.54)	(0.37)	(0.57)	(0.59)	(0.79)	(0.78)	(0.54)	(0.30)	(0.17)	(0.17)	(0.41)	(0.45)	(0.28)	(0.18)
56	0.06	0.06	0.09	0.09	0.10	0.09	0.06	0.06	0.24	0.33	0.26	0.28	0.11	0.09	0.17	0.17
	(0.70)	(0.65)	(0.50)	(0.52)	(0.62)	(0.64)	(0.83)	(0.86)	(0.35)	(0.31)	(0.20)	(0.23)	(0.28)	(0.51)	(0.33)	(0.22)
41	0.06	0.05	0.08	0.11	0.09	0.09	0.07	0.08	0.23	0.25	0.25	0.24	0.12	0.11	0.20	0.21
	(0.70)	(0.63)	(0.62)	(0.39)	(0.58)	(0.65)	(0.76)	(0.76)	(0.49)	(0.34)	(0.11)	(0.15)	(0.39)	(0.49)	(0.20)	(0.21)
22	0.10	0.07	0.12	0.12	0.15	0.12	0.15	0.11	0.30	1.05	0.30	0.26	0.17	0.13	0.28	0.27
	(0.66)	(0.62)	(0.70)	(0.51)	(0.62)	(0.67)	(0.69)	(0.72)	(0.46)	(0.27)	(0.24)	(0.28)	(0.65)	(0.71)	(0.28)	(0.25)
32	0.06	0.05	0.13	0.15	0.10	0.09	0.07	0.08	0.22	0.27	0.29	0.26	0.14	0.13	0.23	0.26
	(0.45)	(0.45)	(0.43)	(0.28)	(0.52)	(0.66)	(0.75)	(0.66)	(0.44)	(0.28)	(0.13)	(0.14)	(0.29)	(0.40)	(0.24)	(0.08)
30	0.06	0.06	0.11	0.12	0.11	0.10	0.11	0.10	0.20	0.25	0.25	0.23	0.14	0.12	0.20	0.24
	(0.67)	(0.54)	(0.69)	(0.20)	(0.61)	(0.66)	(0.80)	(0.65)	(0.47)	(0.27)	(0.20)	(0.16)	(0.36)	(0.34)	(0.24)	(0.06)
27	0.04	0.04	0.06	0.06	0.06	0.07	0.04	0.06	0.19	0.20	0.23	0.19	0.09	0.07	0.16	0.18
	(0.73)	(0.74)	(0.71)	(0.58)	(0.60)	(0.68)	(0.87)	(0.76)	(0.51)	0.43)	(0.13)	(0.27)	(0.37)	(0.57)	(0.28)	(0.18)

Table 7: Benchmarking Analysis results GATE: RMSE and (Coverage)

Notes: This table summarizes the RMSEs and the corresponding coverage rates in parenthesis of all methods when estimating the GATEs, with 100 replications per knob. We refer to this as the Benchmarking Analysis, whereby knob 28 is taken as benchmark and all other knobs vary each criteria one at a time, as shown in Table 6.

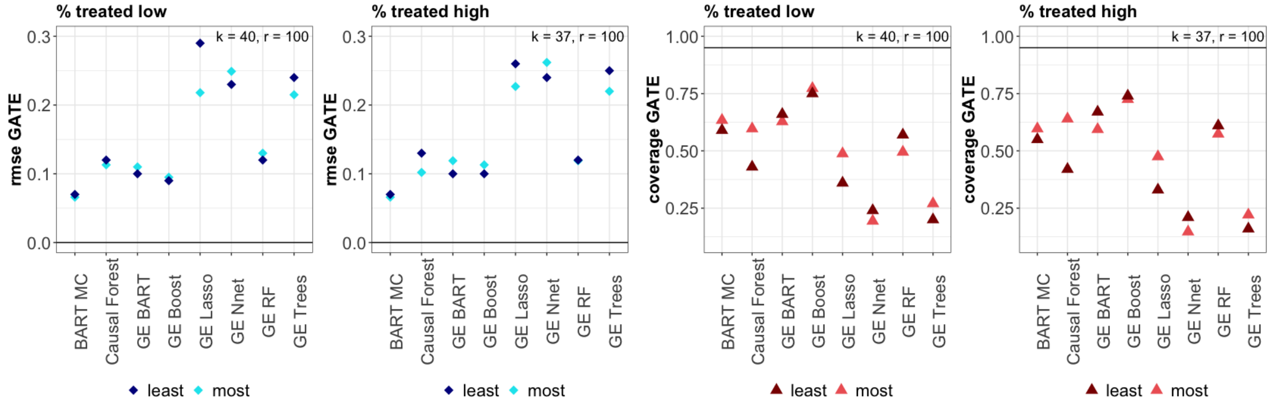
When varying each criteria gradually, the results are mostly similar to the ATE Benchmarking analysis. Thus, we refer for detailed discussion to Section 5.1. By altering the *treatment* model from polynomial to step (knob 56), we observe that the GE Lasso becomes worse, whereas the Tree-based methods slightly improve, especially with respect to coverage. Interestingly, when we change the *percentage of treated* from low to high (knob 41), the performance gap between the most and least affected group in the Causal Forest becomes wider. In particular, as evident in Figure 10, the most affected group outperforms the least affected group considerably, both with respect to RMSE and coverage.



(a) RMSE (b) Coverage

Figure 9: Trt. model polynomial/step

Note: This figure shows the RMSEs and coverage rates of the least and most affected groups in the GATEs estimation, where results are averaged across all knobs with a polynomial *treatment* model versus all knobs with a step *treatment* model. r denotes the number of replications per simulation knob and k is the number of knobs with a given *treatment* model type. As the sum of both ($k = 39$ and $k = 32$) is below the total number of knobs (77), there are remaining knobs with a different *treatment* model, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the RMSE and the coverage rate.

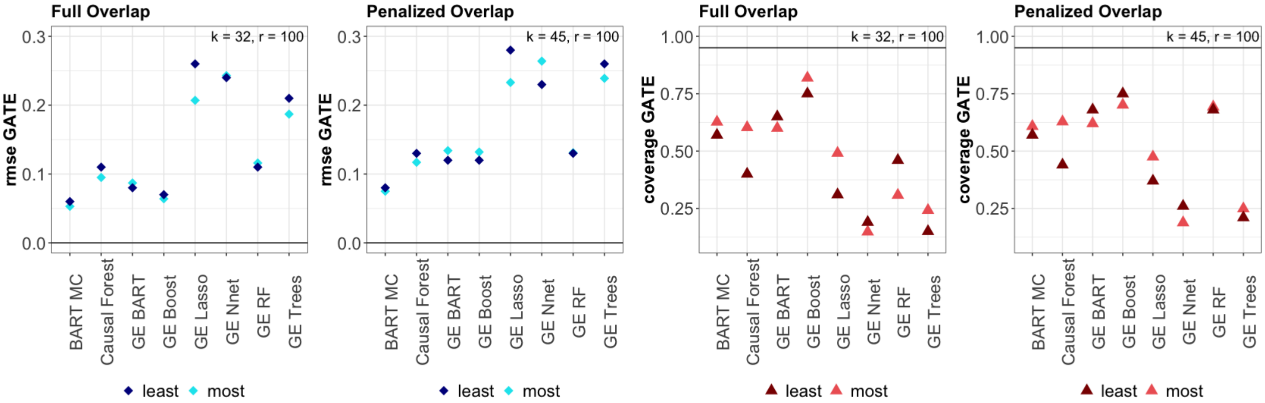


(a) RMSE (b) Coverage

Figure 10: Low/High % treated

Note: This figure shows the RMSEs and coverage rates of the least and most affected groups in the GATEs estimation, where results are averaged across all knobs with low *% treated* versus all knobs with a high *% treated*. r denotes the number of replications per simulation knob and k is the number of knobs with a given *% treated*. The horizontal lines reflect the desired theoretical values of the RMSE and the coverage rate.

When violating the *overlap* assumption (knob 22), GE Boosting clearly deteriorates in performance, as for both groups the RMSE increases substantially and the coverage rates decrease as well. The RMSEs of all the other Generic ML methods also increase, whereas we observe a mixed development in the coverage rates. Surprisingly, some methods have higher coverage, in particular the one of the GE Random Forest increases from 0.41 to 0.65 for the most affected group, and from 0.45 to 0.71 for the least one. However, at the same time we observe an increase in uncertainty, as the interval lengths become much wider (most: from 0.15 to 0.28, least: from 0.15 to 0.26, in Appendix Table 11). The overview over all knobs in Figure 11 shows that GE Boosting and the other Generic ML methods become worse with penalized *overlap*, whereas BART MChains and Causal Forest remain more stable. Figure 25 in Appendix A.4.2 also confirms that the increase in coverage of methods such as GE Random Forest comes at the cost of a substantial increase of the confidence interval lengths.



(a) RMSE

(b) Coverage

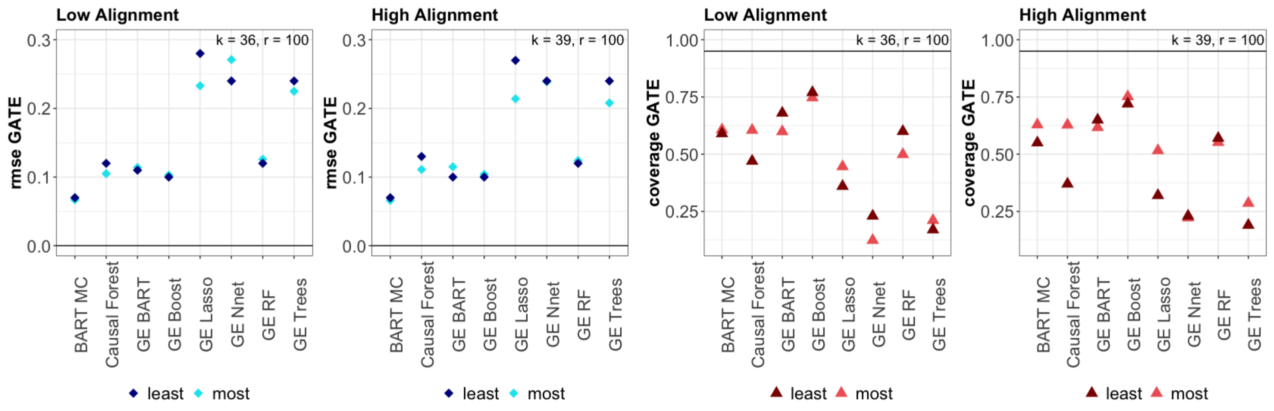
Figure 11: Full/Penalized overlap

Note: This figure shows the RMSEs and coverage rates of the least and most affected groups in the GATEs estimation, where results are averaged across all knobs with full *overlap* versus all knobs with penalized *overlap*. r denotes the number of replications per simulation knob and k is the number of knobs with a given *overlap*. The horizontal lines reflect the desired theoretical values of the RMSE and the coverage rate.

Changing the level of *alignment* from low to high (knob 30) increases the RMSEs of GE BART, GE Boosting and GE Random Forest for both groups, shown in Table 7. More remarkably, most of the coverage rates of the least affected group drop substantially, which also becomes evident in Figure 12. When changing the level of *heterogeneity* from high to low (knob 27), all methods improve with respect to RMSE and also mostly in coverage. Figure 13 further demonstrates the higher RMSEs of all methods with high heterogeneity across all knobs. Intuitively, it implies that the higher the level of heterogeneity in the data, the more challenging it is for all methods to capture this heterogeneity properly. Given that heterogeneous effects are likely to be present in applied research (both in randomized control trials and observational studies), we recommend to estimate GATE by using Tree-based ensemble Causal ML methods (BART MChains, Causal Forest, GE BART, GE Boosting) as they can deal with high-dimensional data and model heterogeneity by construction, so that no prior knowledge or assumptions are required.

Lastly, to evaluate whether the Benchmarking Analysis results of the Generic Boosting, Lasso, Neural Net, Random Forest and Tree are robust given the ML input parameters in Appendix A.6.6 Table 17, we conduct a sensitive check by tuning these parameters over a grid, see Appendix A.7 for more details. Comparing the Benchmarking Analysis GATE results (Table 7 and Appendix A.4.2 Table 11) with the results of the sensitivity check (Appendix A.7 Table 25 and Appendix A.7 Table 26), we observe that the results are robust, indicating that the chosen ML input parameters seem to be representative[9].

[9] Since tuning the ML input parameters over a grid search is computationally much more intensive, we implement the sensitivity analysis only for the knobs of the GATE Benchmarking Analysis and not for the all 77 knobs of the competition.

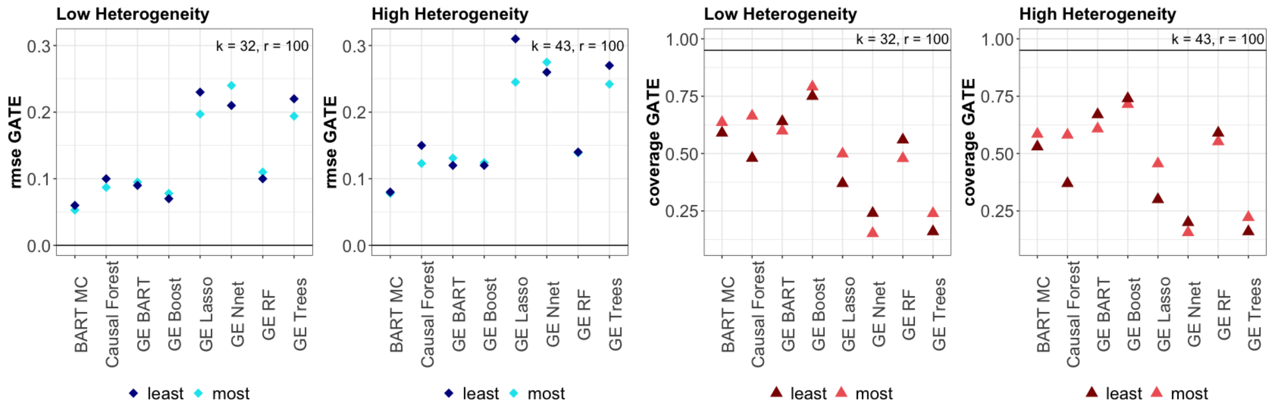


(a) RMSE

(b) Coverage

Figure 12: Low/High alignment

Note: This figure shows the RMSEs and coverage rates of the least and most affected groups in the GATEs estimation, where results are averaged across all knobs with low *alignment* versus all knobs with high *alignment*. r denotes the number of replications per simulation knob and k is the number of knobs with a given level of *alignment*. As the sum of both ($k = 36$ and $k = 39$) is below the total number of knobs (77), there are remaining knobs with a different *alignment* type, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the RMSE and the coverage rate.



(a) RMSE

(b) Coverage

Figure 13: Low/High heterogeneity

Note: This figure shows the RMSEs and coverage rates of the least and most affected groups in the GATEs estimation, where results are averaged across all knobs with low *heterogeneity* versus all knobs with high *heterogeneity*. r denotes the number of replications per simulation knob and k is the number of knobs with a given level of *heterogeneity*. As the sum of both ($k = 32$ and $k = 43$) is below the total number of knobs (77), there are remaining knobs with a different *heterogeneity* type, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the RMSE and the coverage rate.

5.2.2 Individual Treatment Effects

In the following, we evaluate the results of estimating the Individual Treatments Effects in terms of the Precision in Estimating the Heterogeneous Effects (PEHE), see Section 4.2. Figure 14 gives an overview of the PEHE averaged across all simulation knobs. We observe that BART MChains is the best performing method, followed by DR MOM Boosting, DR MOM BART, Causal Forest, DR MOM Random Forest, DR MOM Trees, DR MOM Lasso and DR MOM Neural Net. For comparison, Figure 17 in Appendix A.2 shows the results of the original competition, where Dorie et al. (2019) demonstrate that the Vanilla BART is best suited to estimate ITEs, as it results in lowest PEHE.

Moreover, we can observe that the order of performance of the ML methods within the DR MOM is in-line with the estimation of the nuisance parameter within the DML and Generic framework (Figure 2). Lastly, it is worth mentioning that the Causal Forest is the only method with derived asymptotic results and valid confidence intervals on an individual level, making this method very applicable when the researcher's main objective is inference of Heterogeneous Treatment Effects on a more granular level (Athey et al., 2019).

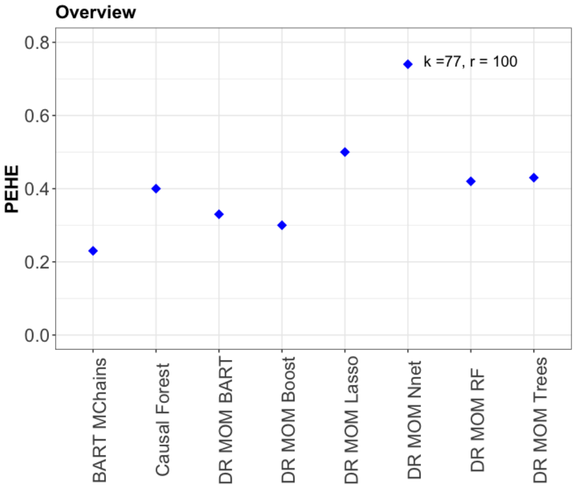


Figure 14: PEHE overview

Note: This figure shows the Precision in Estimating Heterogeneous Effects (PEHE) of all methods in estimating the ITEs, with results averaged across all 77 simulation knobs (k), with 100 replications (r) per knob. Thus, in total we consider 7700 datasets.

Figure 28 in Appendix A.4.3 provides an overview of the PEHE when varying certain criteria levels. We can mostly observe a similar pattern of results as in the ATE and GATE Benchmarking Analysis, as the PEHE slightly increases when the *overlap* assumption is violated or when the level of *heterogeneity* increases. However, the differences overall are less pronounced and evident.

Summarizing, the proposed Causal ML methods can capture heterogeneity both in a form of grouped as well as individual effects. Individual effects could be meaningful in the field of personalized marketing, e.g when analyzing the impact of a change in an online advertising campaign on the user's behaviour, or in the field of personalized medicine, e.g. when analyzing the impact of a new drug on each patient individually. However, researchers are generally more often interested in the effects of a policy or an experiment on certain segments rather than the effects on individuals. For instance, it is not feasible to give tailored recommendations for every participant of an unemployment training, but it is common practise to provide information for a group of participants. Thus, we see higher practical relevance for the methods targeting grouped effects.

6 Empirical application

In the following, we apply the Causal ML methods on an empirical application by analyzing the General Social Survey (GSS). The primary goal is to focus on one of the criteria of the competition and to compare how the methods perform in practice. We choose the criteria *heterogeneity*, because it is present in almost every applied research and its bears high potential for practitioners to extract new insights.

The GSS gathers opinions of Americans regarding national spending behaviour, aiming to provide politicians and policymakers with an unbiased perspective of the society's thoughts[10]. The survey contains a wording experiment, by randomly assigning respondents to answer the same question, but exchanging the term *welfare* with *assistance to the poor*[11]. This experiment has been conducted because several scholars have observed that the term *welfare* is publicly tied to stereotypes such as laziness and waste, instead of its actual meaning to support those in need, and that this negative association is partially driven by racial attitudes (Smith, 1987; Williamson, 1974; Wright, 1977). By applying BART, Green and Kern (2012) detect heterogeneous effects in the GSS (data until 2010), in particular a stronger disregard towards *welfare* by conservative respondents with more explicit radical beliefs, as well as a variation of the effects across the years.

In the following, we employ the considered Causal ML methods of this research on the latest version of the GSS (data until 2018), firstly to analyze whether all methods detect heterogeneity, and secondly to investigate whether the most recent time period provides new insights. We define the response (too much versus not enough spending) as binary outcome variable, *welfare* as treatment group and *assistance to the poor* as control group. Additionally, we choose nine control variables, consisting of questions regarding the attitude towards blacks, age (in years), education (in years), political party identification and liberal-conservative self-placement, see Appendix A.5 Table 12 for an overview of all variables. The sample size consists of 24131 observations.

Due to the random assignment of the treatment, the ATE analysis with Causal ML methods is somewhat trivial, as the propensity score is constant (thus its estimation is not required) and it is theoretically not necessary to condition on a set of covariates to omit hidden bias (Angrist, Oreopoulos, & Williams, 2014). Table 13 in Appendix A.5 shows that nearly all Causal ML methods are identical to the difference in means ATE estimate of 0.35 with a 95% confidence interval of (0.34, 0.36), highlighting the credibility of the Causal ML methods. The ATE estimate can be interpreted as the differences between the proportion of respondents stating that the spending on *welfare* (treatment) is too high and the proportion of respondents stating that the spending on *assistance to the poor* (control) is too

[10] For more information visit https://gss.norc.org, where the data is publicly available.

[11] Exact question: "We are faced with many problems in this country, none of which can be solved easily or inexpensively. I'm going to name some of these problems, and for each one I'd like you to tell me whether you think we're spending too much money on it, too little money, or about the right amount. Are we spending too much, too little, or about the right amount on *welfare/assistance to the poor*?" (*The General Social Survey*, n.d.)

high.

A heterogeneity analysis is of high practical relevance in randomized experiments. The heterogeneity test of the BLP estimator in Appendix A.5 Table 14 detects heterogeneity for all methods. Accordingly, the GATE estimates in Table 15 in Appendix A.5 reveal that a much larger proportion of respondents in the most affected group (compared to the least affected group) thinks spending on *welfare* is too high versus spending on *assistance to the poor*. Interestingly, in contrast to the simulation study, most of the Causal ML methods provide rather similar estimates and confidence intervals, which might be the case because the empirical application is less complex due to the random treatment assignment. For instance, given the constant propensity score, the application does not include complex criteria such as limited *overlap*, high *alignment* or non-linear *treatment* functions. The distributions of the ITEs estimated by BART MChains and Causal Forest in Appendix A.5 Figure 29 also confirms a high level of heterogeneity[12], as the ITEs are ranging from 0.1 up until 0.6, which was to be expected as people have different opinions.

We investigate which covariates[13] are driving the *heterogeneity* by calculating the Conditional Average Treatment Effects (CATEs) of specific covariate levels[14] of the methods providing treatment effects up until the individual level. Table 16 in Appendix A.5 shows that the negative perception of the word *welfare* increases with respondents' negative attitude towards blacks. For instance, the effect is much lower (BART MChains: 0.27, Causal Forest: 0.27) for respondents who think that job inequality between blacks and whites is due to discrimination, whereas it is significantly higher for those who negate this question (BART MChains: 0.4, Causal Forest: 0.4). Furthermore, the most recent GSS data provides interesting insights on the development of the wording effect over the last years. Namely, the perception that spending is too high on *welfare* versus *assistance to the poor* has risen during the years 2010-2014, which is parallel to the Obama administration (see Figures 30). In-line, several scholars have pointed out that the increase in welfare spending has been one of the most controversial elements of the Obama presidency (Conley, 2017). Similar to the results of Green and Kern (2012), Figures 31 and 32 reveal that the opinion of too high spending on welfare is more common across respondents with more conservative political views and Republican party membership, as opposed to to respondents with more liberal views and Democratic party membership.

[12] We omitted the DR MOM methods due to the lacking methodology on uncertainty and confidence intervals, which is crucial in empirical analysis.

[13] In case a large amount of covariates is available, we recommend to narrow down the analysis by utilizing the variable selection procedures within the Causal ML methods. For instance, practitioners can extract the most important variables contributing to the splits of the Causal Forest (see the *variable importance* function in the *grf* R-package for more details).

[14] For instance, if x_D is the observation set of all Democrats, we calculate $\tau_0(x) = \mathbb{E}[Y_i(1) - Y_i(0)|X_i = x_D] = \mathbb{E}[Y_i(1)|X_i = x] - \mathbb{E}[Y_i(0)|X_i = x_D]$.

7 Conclusion

In this research, we investigate the performance of promising Causal ML methods in estimating treatment effects of different granularity by revisiting the 2016 Atlantic Causal Inference Competition, which aims to represent a realistic observational study (Dorie et al., 2019). When estimating the ATE, the best methods perform well across all 77 simulation settings, with an RMSE below 0.05. However, no method reaches nominal coverage. The top performing method is BART MChains, which confirms its strong performance in practice. We find substantial differences within the DML framework, depending on the ML method used as meta-learner. The new proposed DML BART is the second best performing method overall. As the DML framework guarantees convergence in polynomial time and approximately normally distributed estimators, we recommend this estimator if the researcher requires rigorous statistical analysis and hypothesis testing. After DML BART, the best performing ML methods used within the DML are Boosting, Random Forest, Trees, Neural Net and Lasso (in decreasing order), whose order can be explained by the predictive performance of the nuisance parameters. Although the DML framework reveals marginally better results, the estimates of the Generic framework are in a similar range, indicating that its ATE can be used as a side-product when this method is used to analyze heterogeneity. The ATE of the Causal Forest is not among the best performing methods, especially with respect to coverage, indicating that this approach is rather tailored to capture heterogeneity. Lastly, all Causal ML methods outperform the OLS estimator, demonstrating that due to their flexibility, they bear great potential for researchers in settings where traditional estimation techniques would be misspecified (e.g. high-dimensional confounding effects, non-linearities or high heterogeneity).

Analyzing the specific impact of each criteria, by gradually varying the *response/treatment* function, the *percent of treated*, the *overlap* assumption, the *alignment* and the level of *heterogeneity*, provides valuable insights that are not as transparent as when the performance is evaluated in aggregate over all simulation knobs. The Causal ML methods clearly perform best when the functional form of the *treatment* and *response* function matches the merits of the statistical learning technique. For instance, we observe that the Lasso-based methods work better with polynomial than with step functions. When changing the *percent of treated* from low to high, we do not notice any substantial differences. This is reassuring for practitioners as it can be costly or infeasible to obtain a fair 50-50 split. With full *overlap*, we observe that both BART MChains as well as the new proposed DML BART perform exceptionally well, as both reach nominal coverage. When violating the *overlap* assumption, we clearly observe that all Causal ML methods become substantially worse, since the RMSEs increase and the coverage rates drop considerably. However, we notice that BART MChains is less severely affected than the best performing methods of the DML and Generic framework. BART MChains addresses the limited *overlap* by trimming observations based on information from the posterior distribution of the response function, whereas the DML and Generic framework use the propensity score. Given the results, we recommend to use methods that spot areas with limited *overlap* by using information from the response function and omit observations based on data-driven thresholds. Proceeding with the next criteria, we see that the coverage rates of the Causal ML methods decrease with high *alignment*. This is surprising, as Causal ML methods should be able to detect the true confounders of each function due to two separate estimation steps. When increasing the level of *heterogeneity*, we find that also the ATE estimation becomes more challenging. To address this higher complexity, we recommend to use Tree-based ensemble methods, since they model complex interactions by default.

Detecting Heterogeneous Treatment Effects is of high interest in applied econometric research. When estimating the most and the least affected GATEs, we find that all methods perform substantially worse than in the ATE estimation, as the RMSEs of the best performing methods are higher and the coverage rates are far below nominal coverage. Among all methods, the Generic Boosting shows the lowest bias and highest coverage, while BART MChains has the lowest RMSE and interval length, reflecting that ensemble methods based on a weak learner approach seem most suitable. Moreover, we observe that the Causal Forest performs relatively better than in the ATE analysis, highlighting that this method is tailored to capture heterogeneity. Analyzing the impact of the each criteria separately reveals mostly similar insights to the ATE analysis. For instance, limited *overlap* and high *heterogeneity* make the estimation more challenging and require more robust methods that address these complexities to the best extent. When estimating Individual Treatment Effects, we find that BART MChains outperforms all methods, followed by DR MOM Boosting, DR MOM BART, Causal Forest and DR MOM Random Forest, demonstrating that it can be beneficial to extend the set of ML-methods used as meta learner by Knaus et al. (2018), who only use Random Forest and Lasso. Summarizing, across all aggregation levels of treatment effects, the Causal ML methods which model the response and the treatment function flexibly, by growing a sequence of trees based on a weak learner approach, perform best.

Subsequently, we apply the Causal ML method on an empirical application to compare how they are capturing heterogeneity in practice. Specifically, we explore the perception of public spending on *welfare* versus *assistance to the poor*, by using the most recent data of the GSS. We find that all Causal ML method consistently detect positive heterogeneous effects. This implies that in general more respondents think that the spending on *welfare* is too high as opposed to *assistance to the poor*, however the magnitude varies with the respondents' characteristics. BART MChains, Causal Forest and the Generic methods show significant positive point estimates for the most and the least affected GATEs. Moreover, the distribution of the ITEs estimated by the Causal Forest and BART MChains also confirms the high level of heterogeneity. When analyzing the heterogeneity with respect to specific control variables, we find an increase in the effect from the years 2010 until 2014, while the most recent years show a decline. Furthermore, in-line with existing findings of previous versions of the survey, we observe a stronger disdain against *welfare* of respondents with a negative attitude towards blacks and with more conservative views and Republican party membership. In contrast to the simulation study, the estimates of all Causal ML methods are more similar, which is likely as the empirical application has less complex criteria due to the random treatment assignment.

This research provides important and promising insights about the Causal ML methods, but it also leaves room for future research. From a theoretical point of view, only the Causal Forest provides asymptotic theory to form confidence intervals up until the individual level. Thus, due to the strong performance of DR MOM, we see great potential in providing theoretical results for these methods as well. Similarly, theoretical results ensuring that the BART MCMC sampler convergence in polynomial time and that the posterior distribution is concentrated around the true mean are of high interest. From a methodological point of view, more tailored implementations, by using a more extended grid search to tune all input parameters of each ML method at the sample splitting step, or generally increasing the number of splits within the DML and Generic framework, could improve the results. Additionally, it might be worth considering other ML techniques, such as Support Vector Machines or Deep Learning approaches. From a simulation point of view, the impact of *alignment* is of considerable interest, as surprisingly in this research the coverage rates of the Causal ML methods are not stable. Additionally, even if there are already existing techniques, advances in identifying areas of lacking *overlap* would be beneficial for practitioners to satisfy common support. From an empirical point of view, directly comparing the estimates of the traditional methods to the Causal ML methods could access whether the estimates of traditional methods are reliable in the first place. Even more so, by using the recommendations of this research, we are optimistic that the Causal ML methods will provide new valuable insights in applied econometric work.

References

Albert, J. H., & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, *88*(422), 669–679.

Angrist, J., Oreopoulos, P., & Williams, T. (2014). When opportunity knocks, who answers? New evidence on college achievement awards. *Journal of Human Resources*, *49*(3), 572–610.

Athey, S. (2018). The impact of machine learning on economics. In *The economics of artificial intelligence: An agenda.* University of Chicago Press.

Athey, S., & Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, *113*(27), 7353–7360.

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *The Annals of Statistics*, *47*(2), 1148–1178.

Athey, S., & Wager, S. (2019). Estimating treatment effects with causal forests: An application. *arXiv preprint arXiv:1902.07409*.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review*, *78*(1), 1–3.

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

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018a). *Double/debiased machine learning for treatment and structural parameters.* Oxford University Press Oxford, UK.

Chernozhukov, V., Demirer, M., Duflo, E., & Fernandez-Val, I. (2017). *Generic machine learning inference on heterogenous treatment effects in randomized experiments* (Tech. Rep.). cemmap working paper.

Chernozhukov, V., Demirer, M., Duflo, E., & Fernandez-Val, I. (2018b). *Generic machine learning inference on heterogenous treatment effects in randomized experiments* (Tech. Rep.). National Bureau of Economic Research.

Chipman, H. A., George, E. I., McCulloch, R. E., et al. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, *4*(1), 266–298.

Cohen, J. (1962). The statistical power of abnormal-social psychological research: a review. *The Journal of Abnormal and Social Psychology*, *65*(3), 145.

Conley, R. S. (2017). President Obama and the American welfare state: Transformation or punctuation? *PS: Political Science & Politics*, *50*(1), 35–39.

Crump, R. K., Hotz, V. J., Imbens, G. W., & Mitnik, O. A. (2006). *Moving the goalposts: Addressing* *limited overlap in the estimation of average treatment effects by changing the estimand* (Tech. Rep.). National Bureau of Economic Research.

Dorie, V., Hill, J., Shalit, U., Scott, M., Cervone, D., et al. (2019). Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, *34*(1), 43–68.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1) (No. 10). Springer series in statistics New York.

The general social survey. (n.d.). Retrieved 2020-03-29, from `https://gss.norc.org`

Gray, M. W., et al. (1993). Can statistics tell us what we do not want to hear? The case of complex salary structures. *Statistical Science*, *8*(2), 144–158.

Green, D. P., & Kern, H. L. (2012). Modeling heterogeneous treatment effects in survey experiments with Bayesian additive regression trees. *Public opinion quarterly*, *76*(3), 491–511.

Hahn, J. (1998). On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica*, 315–331.

Hainmueller, J., Mummolo, J., & Xu, Y. (2019). How much should we trust estimates from multiplicative interaction models? Simple tools to improve empirical practice. *Political Analysis*, *27*(2), 163–192.

Hill, J., & Su, Y.-S. (2013). Assessing lack of common support in causal inference using Bayesian nonparametrics: Implications for evaluating the effect of breastfeeding on children's cognitive outcomes. *The Annals of Applied Statistics*, 1386–1420.

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, *20*(1), 217–240.

Hothorn, T., Lausen, B., Benner, A., & Radespiel-Tröger, M. (2004). Bagging survival trees. *Statistics in medicine*, *23*(1), 77–91.

Imbens, G. W., & Wooldridge, J. M. (2009). Recent developments in the econometrics of program evaluation. *Journal of economic literature*, *47*(1), 5–86.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Knaus, M., Lechner, M., & Strittmatter, A. (2018). Machine learning estimation of heterogeneous causal effects: Empirical Monte Carlo evidence.

Kuhn, M., et al. (2008). Building predictive models in R using the caret package. *Journal of statistical software*, *28*(5), 1–26.

LaLonde, R. J. (1986). Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, 604–620.

Li, F., Morgan, K. L., & Zaslavsky, A. M. (2018). Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, *113*(521), 390–400.

Nie, X., & Wager, S. (2017). Quasi-oracle estimation of heterogeneous treatment effects. *arXiv preprint arXiv:1712.04912*.

Niswander, K. R. (1972). The collaborative perinatal study of the National Institute of Neurological Diseases and Stroke. *The Woman and Their Pregnancies*.

Ridgeway, G. (2007). Generalized boosted models: A guide to the gbm package. *Update*, *1*(1), 2007.

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

Rosenbaum, P. R. (1987). Model-based direct adjustment. *Journal of the American Statistical Association*, *82*(398), 387–394.

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

Rothe, C. (2017). Robust confidence intervals for average treatment effects under limited overlap. *Econometrica*, *85*(2), 645–660.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, *66*(5), 688.

Schiltz, F., Masci, C., Agasisti, T., & Horn, D. (2018). Using regression tree ensembles to model interaction effects: a graphical approach. *Applied Economics*, *50*(58), 6341–6354.

Sexton, J., & Laake, P. (2009). Standard errors for bagged and random forest estimators. *Computational Statistics & Data Analysis*, *53*(3), 801–811.

Słoczyński, T. (2015). New evidence on linear regression and treatment effect heterogeneity.

Smith, T. W. (1987). That which we call welfare by any other name would smell sweeter an analysis of the impact of question wording on response patterns. *Public opinion quarterly*, *51*(1), 75–83.

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*(523), 1228–1242.

Williamson, J. B. (1974). Beliefs about the motivation of the poor and attitudes toward poverty policy. *Social Problems*, *21*(5), 634–648.

Wright, G. C. (1977). Racism and welfare policy in America. *Social Science Quarterly*, *57*(4), 718–730.

A Appendices

A.1 Overview simulation knobs by Dorie et al. (2019)

Knob	Treatment Model	% treated	Overlap	Response Model	Alignment	Heterogeneity
1	linear	low	penalize	linear	high	high
2	polynomial	low	penalize	exponential	high	none
3	linear	low	penalize	linear	high	none
4	polynomial	low	full	exponential	high	high
5	linear	low	penalize	exponential	high	high
6	polynomial	low	penalize	linear	high	high
7	polynomial	low	penalize	exponential	high	high
8	polynomial	low	penalize	exponential	none	high
9	step	low	penalize	step	high	high
10	linear	low	penalize	exponential	low	high
11	polynomial	low	penalize	linear	low	high
12	polynomial	low	penalize	exponential	low	high
13	linear	high	penalize	exponential	high	high
14	polynomial	high	penalize	linear	high	high
15	polynomial	high	penalize	exponential	high	high
16	polynomial	high	penalize	exponential	none	high
17	step	high	penalize	step	high	high
18	linear	high	penalize	exponential	low	high
19	polynomial	high	penalize	linear	low	high
20	polynomial	high	penalize	exponential	low	high
21	polynomial	low	penalize	step	low	low
22	polynomial	low	penalize	step	low	high
23	polynomial	low	penalize	step	high	low
24	polynomial	low	penalize	step	high	high
25	polynomial	low	penalize	exponential	low	low
26	polynomial	low	penalize	exponential	high	low
27	polynomial	low	full	step	low	low
28	polynomial	low	full	step	low	high
29	polynomial	low	full	step	high	low
30	polynomial	low	full	step	high	high
31	polynomial	low	full	exponential	low	low
32	polynomial	low	full	exponential	low	high
33	polynomial	low	full	exponential	high	low
34	polynomial	high	penalize	step	low	low
35	polynomial	high	penalize	step	low	high
36	polynomial	high	penalize	step	high	low
37	polynomial	high	penalize	step	high	high
38	polynomial	high	penalize	exponential	low	low
39	polynomial	high	penalize	exponential	high	low
40	polynomial	high	full	step	low	low
41	polynomial	high	full	step	low	high

42	polynomial	high	full	step	high	low
43	polynomial	high	full	step	high	high
44	polynomial	high	full	exponential	low	low
45	polynomial	high	full	exponential	low	high
46	polynomial	high	full	exponential	high	low
47	polynomial	high	full	exponential	high	high
48	step	low	penalize	step	low	low
49	step	low	penalize	step	low	high
50	step	low	penalize	step	high	low
51	step	low	penalize	exponential	low	low
52	step	low	penalize	exponential	low	high
53	step	low	penalize	exponential	high	low
54	step	low	penalize	exponential	high	high
55	step	low	full	step	low	low
56	step	low	full	step	low	high
57	step	low	full	step	high	low
58	step	low	full	step	high	high
59	step	low	full	exponential	low	low
60	step	low	full	exponential	low	high
61	step	low	full	exponential	high	low
62	step	low	full	exponential	high	high
63	step	high	penalize	step	low	low
64	step	high	penalize	step	low	high
65	step	high	penalize	step	high	low
66	step	high	penalize	exponential	low	low
67	step	high	penalize	exponential	low	high
68	step	high	penalize	exponential	high	low
69	step	high	penalize	exponential	high	high
70	step	high	full	step	low	low
71	step	high	full	step	low	high
72	step	high	full	step	high	low
73	step	high	full	step	high	high
74	step	high	full	exponential	low	low
75	step	high	full	exponential	low	high
76	step	high	full	exponential	high	low
77	step	high	full	exponential	high	high

Table 8: Overview of all 77 simulation knobs

A.2 Original results from ACIC competition 2016

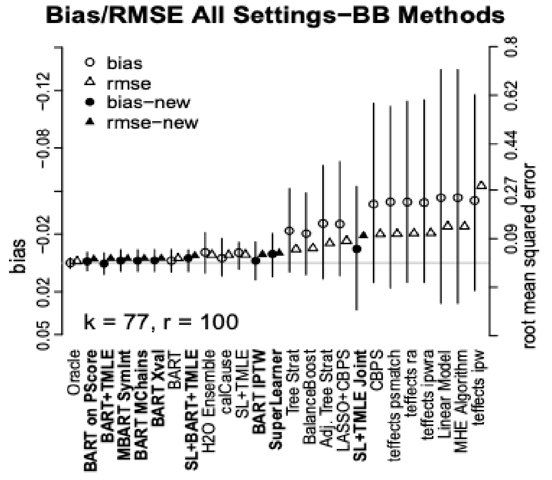


Fig 2: This plot displays both bias (left y-axis) and RMSE (right y-axis) for all submitted black box methods and newly created methods. Both are calculated across the 7700 data sets in the black box competition. Bias is displayed by circles and RMSE by triangles, each averaged across all the data sets; open symbols are used for submitted methods and filled for newly created methods. Lines for bias measures show the interquartile range of all biases across the 77 settings and 100 replications.

Figure 15: Bias/RMSE when estimating ATT by Dorie et al. (2019)

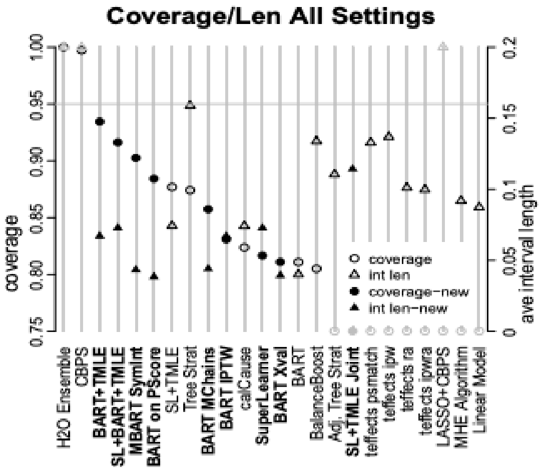


Fig 3: Coverage (circles) and average interval length (triangles) for all of the black box and newly created methods across the 7700 black box data sets. Methods are ordered according to decreasing coverage rates. Methods in bold/filled plot points represent the newly created methods. Points in gray were beyond the plotting region (very poor coverage or very large intervals) and are shown at the corresponding top or bottom edge.

Figure 16: Coverage/Int Length when estimating ATT by Dorie et al. (2019)

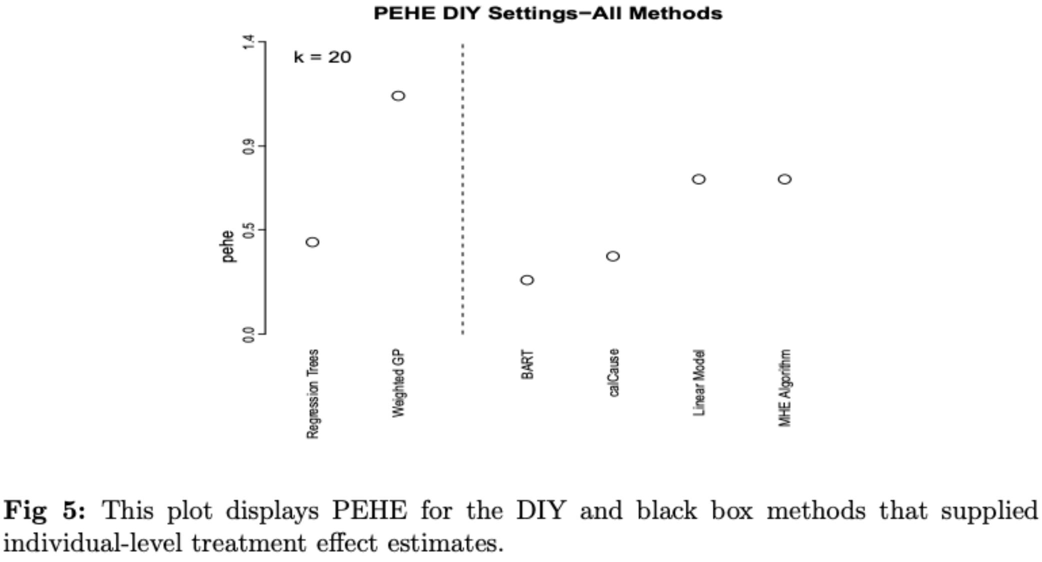


Fig 5: This plot displays PEHE for the DIY and black box methods that supplied individual-level treatment effect estimates.

Figure 17: Estimation of PEHE by Dorie et al. (2019)

Note: This figure shows the PEHE of all the methods in the competition that provided Individual Treatment Effects. The two methods to the left of dotted line were only applied on 20 simulation knobs.

A.3 Supporting material

variable	description	variable	description
x_1	mom__age	x_{29}	dad__years__educ
x_2	mar__status	x_{30}	num__premes
x_3	mom__cigs__per__day	x_{31}	num__abortions
x_4	mom__years__smoked	x_{32}	num__prior__pregs
x_5	mom__height	x_{33}	num__stillbirths
x_6	mom__weight__prior	x_{34}	bayley__mental
x_7	mom__num__cardio__cond	x_{35}	bayley__motor
x_8	mom__num__pulm__cond	x_{36}	placental__weight
x_9	mom__num__hema__cond	x_{37}	cord__length
x_{10}	mom__num__endocrine__cond	x_{38}	sex
x_{11}	mom__num__veneral__cond	x_{39}	apgar__1m__total
x_{12}	mom__num__urin__cond	x_{40}	apgar__5m__total
x_{13}	mom__num__gyne__cond	x_{41}	bottle__feed__days
x_{14}	mom__num__neur__cond	x_{42}	breast__feed__days
x_{15}	mom__num__obst__compl	x_{43}	child__bilirubin
x_{16}	mom__num__infect__dis	x_{44}	child__hematocrit
x_{17}	mom__work__status	x_{45}	child__hemoglobin
x_{18}	mom__years__educ	x_{46}	child__num__neur__abn
x_{19}	family__income	x_{47}	child__num__cns__cond
x_{20}	housing__density	x_{48}	child__num__muscoskel
x_{21}	mom__birth__place	x_{49}	child__num__resp__abn
x_{22}	consanguinity	x_{50}	child__num__cardio__abn
x_{23}	socio__eco	x_{51}	child__num__liver__abn
x_{24}	mom__race	x_{52}	child__num__hemo__cond
x_{25}	age__menarche	x_{53}	child__num__infect
x_{26}	dias__blood__pres	x_{54}	child__num__synd
x_{27}	mom__weight__birth	x_{55}	child__num__endo__dis
x_{28}	dad__age	x_{56}	child__num__proc

Table 9: Overview of 58 control variables that could have been selected in the Collaborative Perinatal Project to capture confounding effects

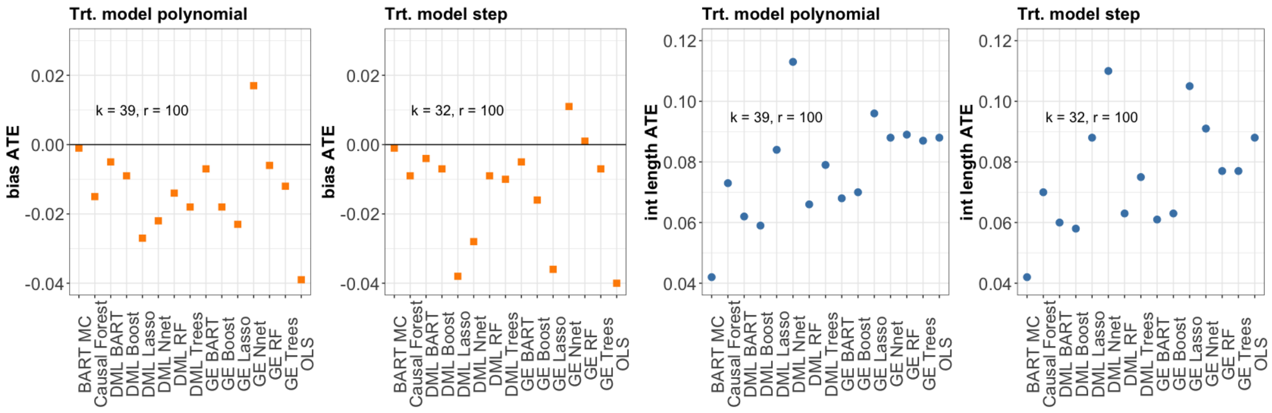
A.4 Additional results

A.4.1 Average Treatment Effect

knob	BART	Causal	DML	DML	DML	DML	DML	DML	GE	GE	GE	GE	GE RF	GE	OLS
	MC	Forest	BART	Boost	Lasso	Nnet	RF	Trees	BART	Boost	Lasso	Nnet		Trees	
27	-0.00 (0.03)	-0.01 (0.06)	-0.00 (0.05)	-0.00 (0.05)	-0.01 (0.08)	-0.01 (0.09)	-0.01 (0.06)	-0.01 (0.06)	0.00 (0.05)	-0.00 (0.05)	-0.01 (0.10)	0.02 (0.08)	0.01 (0.06)	-0.01 (0.07)	-0.02 (0.08)
55	-0.00 (0.03)	-0.00 (0.06)	-0.00 (0.05)	-0.00 (0.05)	-0.01 (0.08)	-0.01 (0.09)	-0.00 (0.05)	-0.00 (0.06)	0.01 (0.04)	-0.00 (0.04)	-0.01 (0.10)	0.02 (0.08)	0.02 (0.05)	0.00 (0.06)	-0.01 (0.08)
40	-0.00 (0.03)	-0.01 (0.06)	-0.00 (0.05)	-0.00 (0.05)	-0.01 (0.08)	-0.01 (0.09)	-0.00 (0.06)	-0.01 (0.06)	0.00 (0.04)	-0.00 (0.04)	-0.01 (0.10)	0.02 (0.08)	0.01 (0.05)	-0.00 (0.07)	-0.01 (0.08)
21	0.00 (0.04)	-0.01 (0.06)	-0.00 (0.05)	-0.01 (0.05)	-0.01 (0.08)	-0.01 (0.12)	-0.00 (0.06)	-0.01 (0.07)	-0.00 (0.06)	-0.02 (0.07)	-0.01 (0.11)	0.03 (0.10)	0.00 (0.07)	-0.00 (0.08)	-0.01 (0.09)
31	0.00 (0.03)	-0.01 (0.07)	-0.00 (0.06)	-0.00 (0.06)	-0.01 (0.07)	-0.01 (0.09)	-0.00 (0.07)	-0.01 (0.07)	0.01 (0.05)	-0.00 (0.05)	-0.01 (0.08)	0.02 (0.08)	0.01 (0.06)	-0.00 (0.08)	-0.01 (0.08)
29	-0.00 (0.03)	-0.01 (0.06)	-0.01 (0.05)	-0.01 (0.05)	-0.04 (0.07)	-0.03 (0.09)	-0.02 (0.06)	-0.03 (0.06)	0.00 (0.05)	-0.01 (0.04)	-0.03 (0.08)	0.01 (0.07)	0.00 (0.06)	-0.02 (0.07)	-0.05 (0.08)
28	-0.00 (0.03)	-0.00 (0.07)	-0.01 (0.06)	-0.01 (0.06)	-0.04 (0.08)	-0.03 (0.10)	-0.02 (0.07)	-0.02 (0.07)	-0.00 (0.05)	-0.01 (0.05)	-0.03 (0.09)	0.01 (0.08)	0.01 (0.06)	-0.01 (0.07)	-0.05 (0.09)

Table 10: Benchmarking Analysis results ATE: Bias and (Int Length)

Notes: This table summarizes the bias and the interval length in parenthesis of all methods when estimating the ATE, with 100 replications per knob. We refer to this as the Benchmarking Analysis, whereby knob 27 is taken as benchmark and all other knobs vary each criteria one at a time, as shown in Table 3.

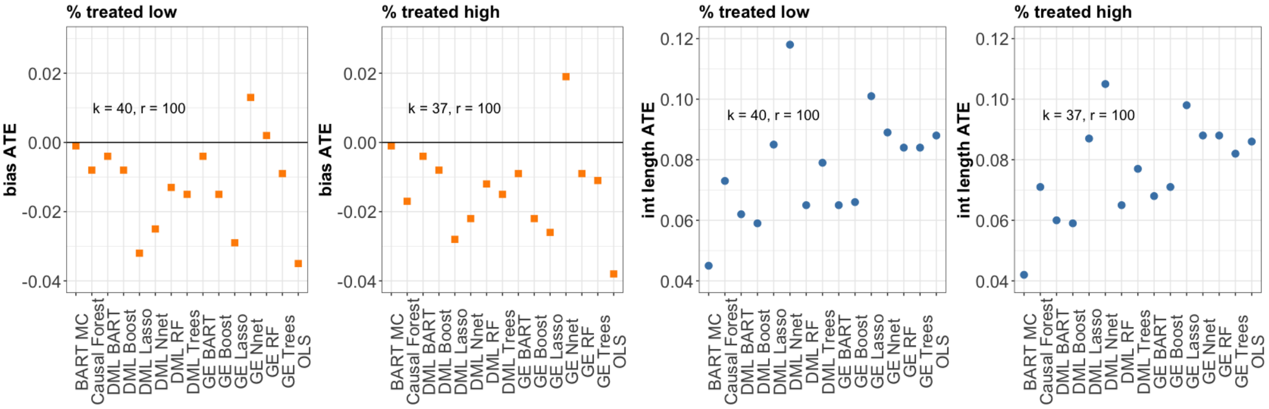


(a) Bias

(b) Interval Length

Figure 18: Trt. model polynomial/step

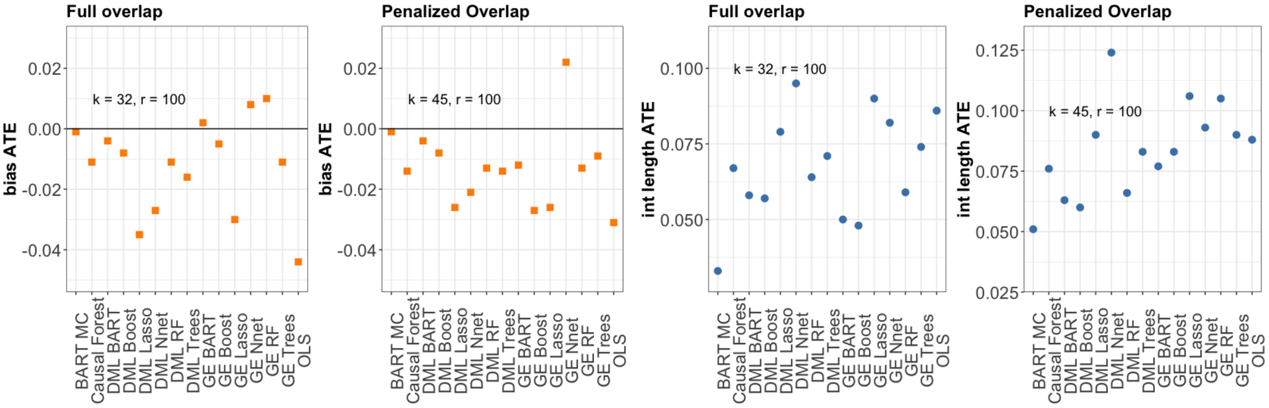
Note: This figure shows the bias and interval length of the ATE estimation, where results are averaged across all knobs with a polynomial *treatment* model versus all knobs with a step *treatment* model. r denotes the number of replications per simulation knob and k is the number of knobs with a given *treatment* model type. As the sum of both ($k = 39$ and $k = 32$) is below the total number of knobs (77), there are remaining knobs with a different *treatment* model, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the bias.



(a) Bias (b) Interval Length

Figure 19: Low/High % treated

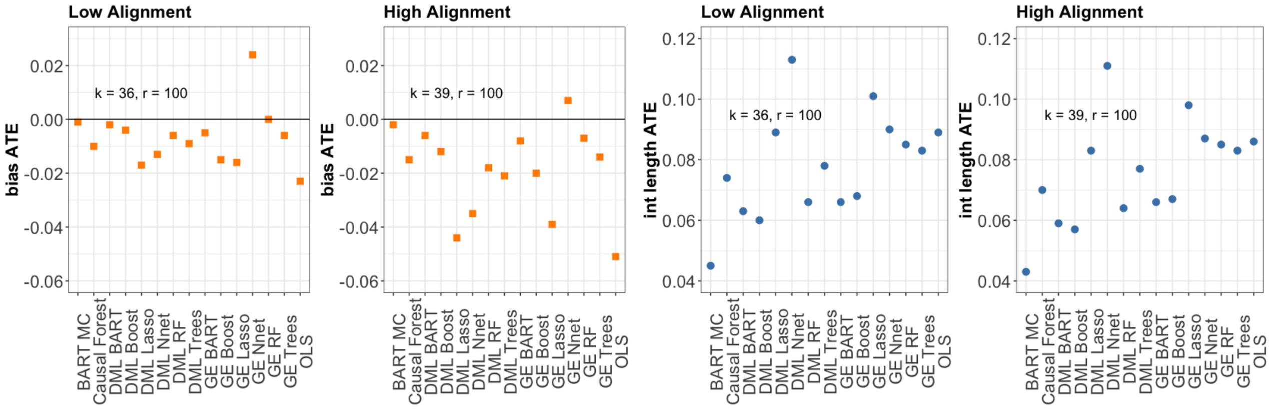
Note: This figure shows the bias and interval length of the ATE estimation, where results are averaged across all knobs with low *% treated* versus all knobs with a high *% treated*. r denotes the number of replications per simulation knob and k is the number of knobs with a given *% treated*. The horizontal lines reflect the desired theoretical values of the bias.



(a) Bias (b) Interval Length

Figure 20: Full/Penalized overlap

Note: This figure shows the bias and interval length of the ATE estimation, where results are averaged across all knobs with full *overlap* versus all knobs with penalized *overlap*. r denotes the number of replications per simulation knob and k is the number of knobs with a given *overlap*. The horizontal lines reflect the desired theoretical values of the bias.

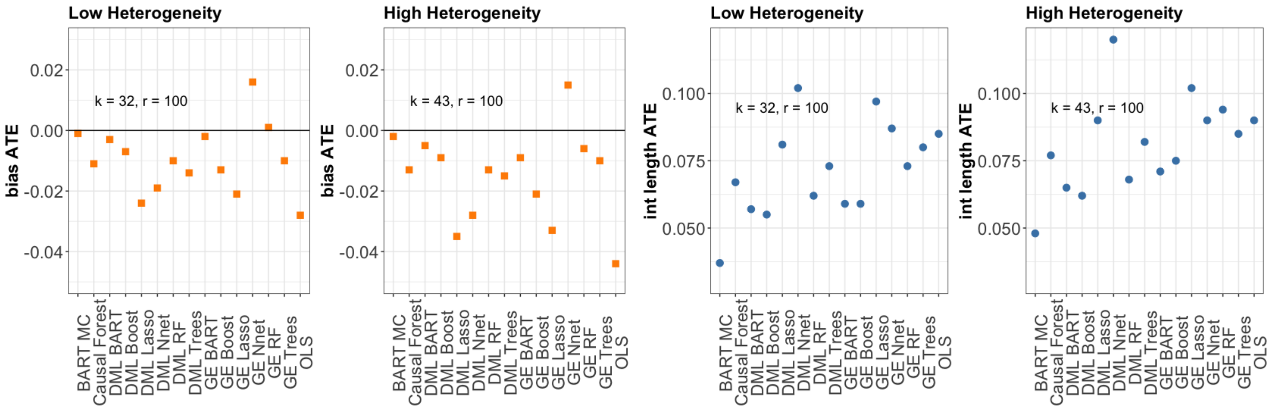


(a) Bias

(b) Interval Length

Figure 21: Low/High alignment

Note: This figure shows the bias and interval length of the ATE estimation, where results are averaged across all knobs with low *alignment* versus all knobs with high *alignment*. r denotes the number of replications per simulation knob and k is the number of knobs with a given level of *alignment*. As the sum of both ($k = 36$ and $k = 39$) is below the total number of knobs (77), there are remaining knobs with a different *alignment* type, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the bias.



(a) Bias

(b) Interval Length

Figure 22: Low/High heterogeneity

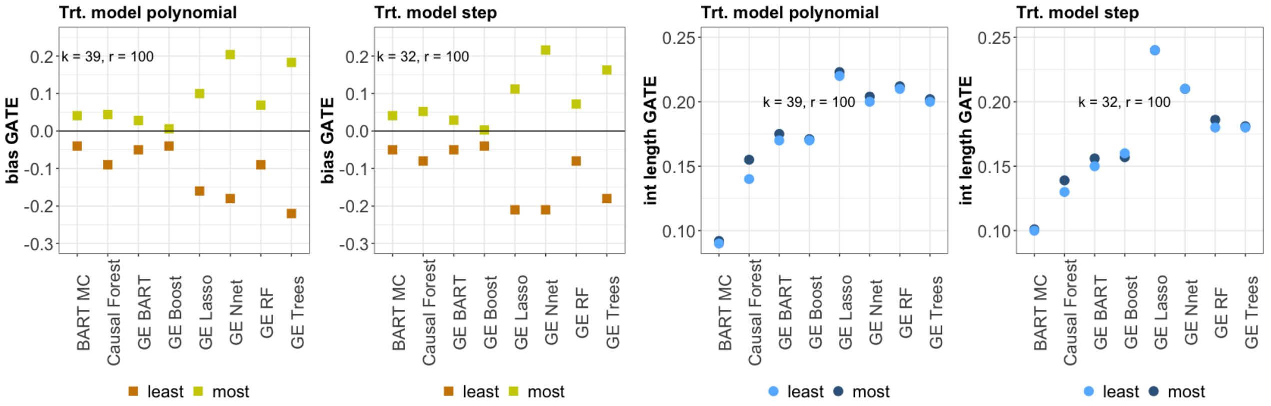
Note: This figure shows the bias and interval length of the ATE estimation, where results are averaged across all knobs with low *heterogeneity* versus all knobs with high *heterogeneity*. r denotes the number of replications per simulation knob and k is the number of knobs with a given level of *heterogeneity*. As the sum of both ($k = 32$ and $k = 43$) is below the total number of knobs (77), there are remaining knobs with a different *heterogeneity* type, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the bias.

A.4.2 Group Average Treatment Effects

knob	BART MC		Causal Forest		GE BART		GE Boost		GE Lasso		GE Nnet		GE RF		GE Trees	
	most	least	most	least	most	least	most	least	most	least	most	least	most	least	most	least
28	0.04 (0.09)	-0.04 (0.09)	0.05 (0.13)	-0.08 (0.13)	0.05 (0.14)	-0.05 (0.14)	0.03 (0.13)	-0.04 (0.12)	0.11 (0.20)	-0.19 (0.20)	0.21 (0.19)	-0.19 (0.19)	0.09 (0.15)	-0.09 (0.15)	0.16 (0.17)	-0.18 (0.17)
56	0.04 (0.10)	-0.04 (0.10)	0.06 (0.12)	-0.06 (0.12)	0.05 (0.12)	-0.04 (0.12)	0.02 (0.11)	-0.03 (0.11)	0.09 (0.24)	-0.22 (0.24)	0.20 (0.20)	-0.21 (0.20)	0.09 (0.13)	-0.07 (0.13)	0.14 (0.15)	-0.14 (0.15)
41	0.04 (0.10)	-0.04 (0.10)	0.05 (0.14)	-0.07 (0.12)	0.05 (0.14)	-0.05 (0.14)	0.04 (0.12)	-0.04 (0.13)	0.13 (0.20)	-0.17 (0.20)	0.22 0.19)	-0.20 (0.19)	0.09 (0.15)	-0.08 (0.15)	0.17 (0.18)	-0.18 (0.18)
22	0.05 (0.13)	-0.05 (0.12)	0.05 (0.17)	-0.08 (0.16)	0.03 (0.21)	-0.05 (0.20)	0.01 (0.20)	-0.04 (0.20)	0.09 (0.31)	-0.30 (0.30)	0.24 0.24)	-0.20 (0.24)	0.09 (0.28)	-0.08 (0.26)	0.21 (0.21)	-0.20 (0.21)
32	0.05 (0.09)	-0.05 (0.08)	0.09 (0.16)	-0.12 (0.14)	0.06 (0.15)	-0.05 (0.15)	0.04 (0.14)	-0.05 (0.14)	0.13 (0.21)	-0.20 (0.21)	0.24 (0.20)	-0.22 (0.20)	0.12 (0.16)	-0.11 (0.16)	0.20 (0.20)	-0.24 (0.20)
30	0.04 (0.09)	-0.05 (0.08)	0.04 (0.14)	-0.10 (0.12)	0.05 (0.14)	-0.06 (0.14)	0.04 (0.13)	-0.06 (0.13)	0.08 (0.19)	-0.19 (0.19)	0.19 (0.18)	-0.19 (0.18)	0.10 (0.15)	-0.10 (0.15)	0.16 (0.18)	-0.22 (0.18)
27	0.03 (0.09)	-0.03 (0.08)	0.03 (0.12)	-0.05 (0.12)	0.04 (0.12)	-0.03 (0.12)	0.02 (0.11)	-0.04 (0.12)	0.12 (0.22)	-0.14 (0.22)	0.20 (0.19)	-0.15 (0.19)	0.08 (0.13)	-0.06 (0.13)	0.13 (0.16)	-0.16 (0.16)

Table 11: Benchmarking Analysis results GATE: Bias and (Int Length)

Notes: This table summarizes the bias and the interval length in parenthesis of all methods when estimating the GATEs, with 100 replications per knob. We refer to this as the Benchmarking Analysis, whereby knob 28 is taken as benchmark and all other knobs vary each criteria one at a time, as shown in Table 6.

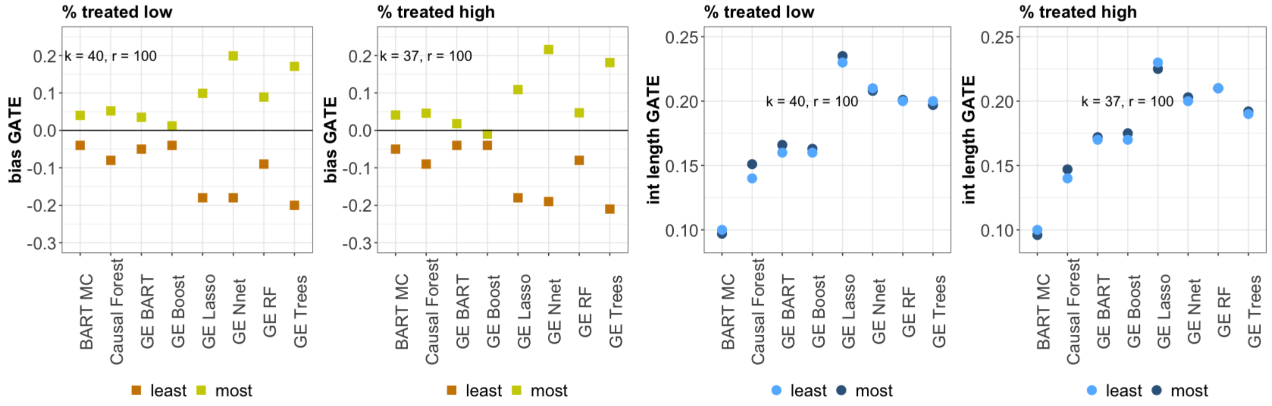


(a) Bias

(b) Interval Length

Figure 23: Trt. model polynomial/step

Note: This figure shows the bias and interval length of the least and most affected groups in the GATEs estimation, where results are averaged across all knobs with a polynomial *treatment* model versus all knobs with a step *treatment* model. r denotes the number of replications per simulation knob and k is the number of knobs with a given *treatment* model type. As the sum of both ($k = 39$ and $k = 32$) is below the total number of knobs (77), there are remaining knobs with a different *treatment* model, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the bias.

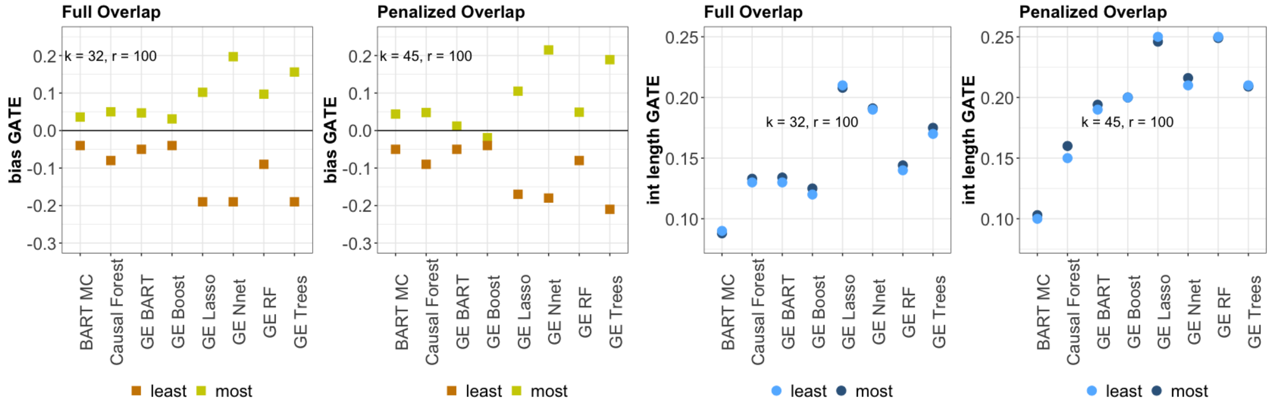


(a) Bias

(b) Interval Length

Figure 24: Low/High % treated

Note: This figure shows the bias and interval length of the least and most affected groups in the GATEs estimation, where results are averaged across all knobs with low *% treated* versus all knobs with a high *% treated*. r denotes the number of replications per simulation knob and k is the number of knobs with a given *% treated*. The horizontal lines reflect the desired theoretical values of the bias.

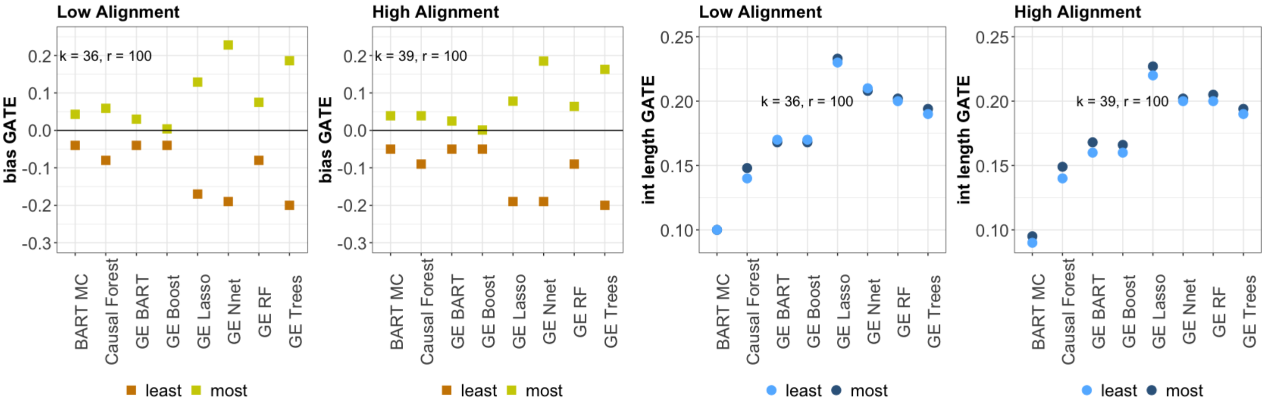


(a) Bias

(b) Interval Length

Figure 25: Full/Penalized overlap

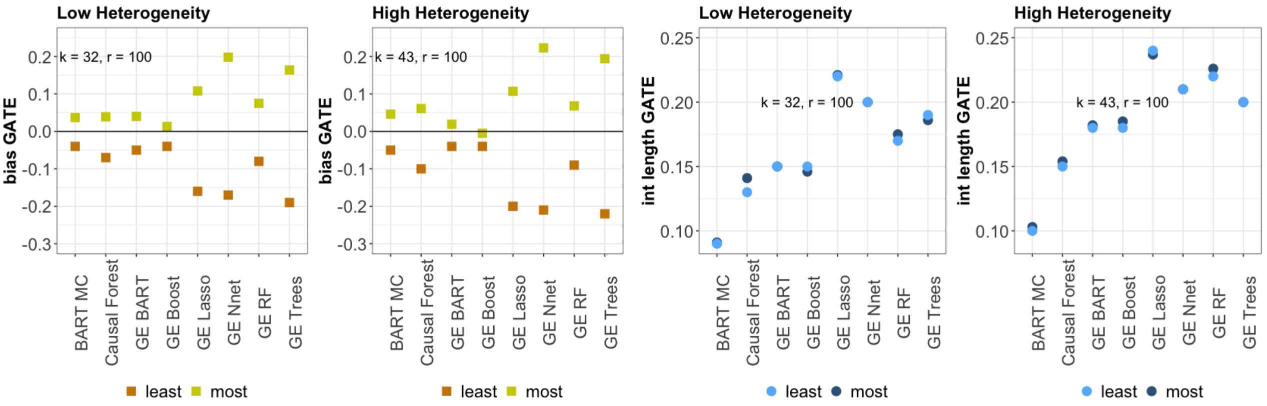
Note: This figure shows the bias and interval length of the least and most affected groups in the GATEs estimation, where results are averaged across all knobs with full *overlap* versus all knobs with penalized *overlap*. r denotes the number of replications per simulation knob and k is the number of knobs with a given *overlap*. The horizontal lines reflect the desired theoretical values of the bias.



(a) Bias (b) Interval Length

Figure 26: Low/High alignment

Note: This figure shows the bias and interval length of the least and most affected groups in the GATEs estimation, where results are averaged across all knobs with low *alignment* versus all knobs with high *alignment*. r denotes the number of replications per simulation knob and k is the number of knobs with a given level of *alignment*. As the sum of both ($k = 36$ and $k = 39$) is below the total number of knobs (77), there are remaining knobs with a different *alignment* type, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the bias.

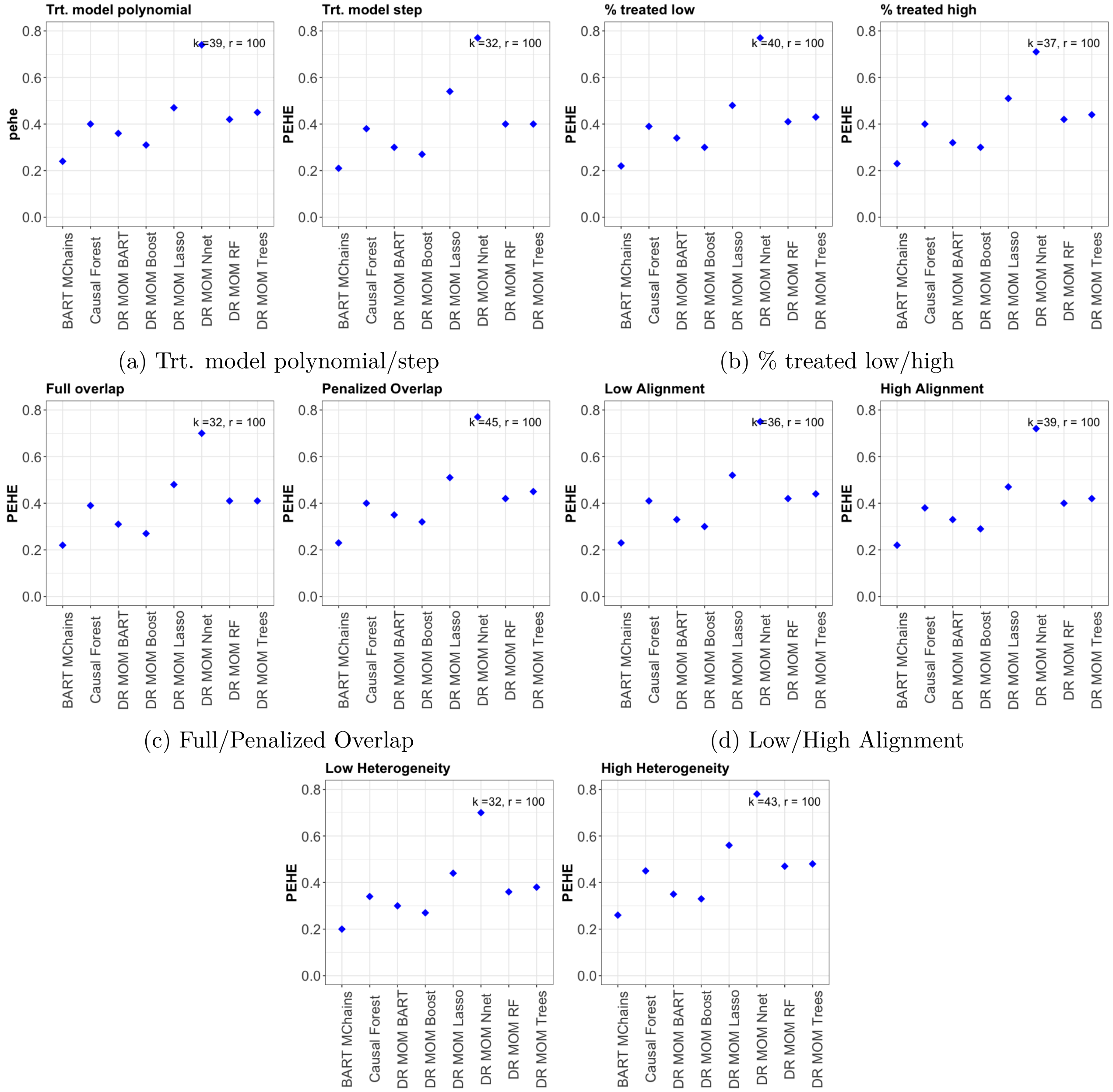


(a) Bias (b) Interval Length

Figure 27: Low/High heterogeneity

Note: This figure shows the bias and interval length of the least and most affected groups in the GATEs estimation, where results are averaged across all knobs with low *heterogeneity* versus all knobs with high *heterogeneity*. r denotes the number of replications per simulation knob and k is the number of knobs with a given level of *heterogeneity*. As the sum of both ($k = 32$ and $k = 43$) is below the total number of knobs (77), there are remaining knobs with a different *heterogeneity* type, see Table 8 for reference. The horizontal lines reflect the desired theoretical values of the bias.

A.4.3 Individual Treatment Effects



(a) Trt. model polynomial/step

(b) % treated low/high

(c) Full/Penalized Overlap

(d) Low/High Alignment

(e) Low/High Heterogeneity

Figure 28: Trt.modeld / %treated / Overlap / Alignment / Heterogeneity

Note: This figure shows the performance of all methods in estimating the ITEs, where results are averaged across all knobs with a given criteria, such as: type of *treatment* model, *% treated*, *overlap* assumption, level of *alignment* and level of *heterogeneity*. In this figure, k refers to the number of knobs with a given criteria and r is the number of simulation replications.

A.5 Results GSS application

variable	**variable information**
response	Answer: too little/about right: (17399); Answer: too much (6732)
treatment	assistance (12199); welfare (11932)
Party identification	NOT STR DEMOCRAT (4722); NOT STR REPUBLICAN (3964); STRONG DEMOCRAT (3813); independent (3342) IND,NEAR DEM (2953); STRONG REPUBLICAN (2625); Other (2712)
age	Min.: 18.00; 1st Qu.: 32.00; Median : 43.00; Mean: 45.94; 3rd Qu.: 58.00; Max.: 89.00
educ	Min.: 0.00; 1st Qu.: 12.00; Median: 13.00; Mean: 13.38; 3rd Qu.: 16.00; Max.: 20.00
Political views	EXTREMELY LIBERAL (823); liberal (2916); SLIGHTLY LIBERAL (3017); moderate (9069); SLGHTLY CONSERVATIVE (3746); conservative (3729); EXTRMLY CONSERVATIVE (831)
year	Min.: 1985; 1st Qu.: 1991; Median: 2000; Mean: 2001; 3rd Qu.: 2010; Max.: 2018
RACDIF1	yes (9664); no (14467)
RACDIF2	yes (3048); no (21083)
RACDIF3	yes (11923); no (12208)
RACDIF4	yes (12237); no (11894)

Table 12: Overview variables GSS data

Notes: This table presents the eleven variables from the GSS survey used to model the effect of the wording *welfare* versus *assistance on the poor* on the perception of spending behavior. The response variable is the answer to the question "We are faced with many problems in this country, none of which can be solved easily or inexpensively (...). Are we spending too much, too little, or about the right amount on *welfare*(treatment)/*assistance to the poor*(control)?" (*The General Social Survey*, n.d.). RACDIF1-RACDIF4 are questions indicating negative attitude towards blacks, see Table 16 for more details. Parenthesis denote the number of observations of a level in a categorical variable. Numerical variables are summarized by the minimum, mean, maximum and the corresponding quantiles. The total number of observations is 24131.

method	DML BART	DML Boosting	DML Forest	DML Lasso	DML Nnet	DML Trees
ATE	0.35	0.35	0.35	0.35	0.35	0.35
CI	(0.34, 0.36)	(0.34, 0.36)	(0.34, 0.36)	(0.34, 0.36)	(0.34, 0.36)	(0.34, 0.36)
method	GE BART	GE Boosting	GE Forest	GE Lasso	GE Nnet	GE Trees
ATE	0.35	0.35	0.32	0.35	0.35	0.35
CI	(0.33,0.36)	(0.33,0.36)	(0.31,0.34)	(0.33,0.36)	(0.33,0.36)	(0.34,0.36)
method	BART MC	CF				
ATE	0.35	0.35				
CI	(0.34, 0.36)	(0.34, 0.36)				

Table 13: Results of the ATE estimation on the GSS

Notes: The ATE estimates the differences between the proportion of respondents stating that the spending on *welfare* (treatment) is too high and the proportion of respondents stating that the spending on *assistance to the poor* (control) is too high. CI denote the confidence intervals. The DML and Generic frameworks are implemented with 100 splits.

	Causal Forest	GE BART	GE Boosting	GE Forest	GE Lasso	GE Nnet	GE Trees
BLP(β_1)	1.19	0.83	0.94	0.38	0.75	0.67	0.55
p-value	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)

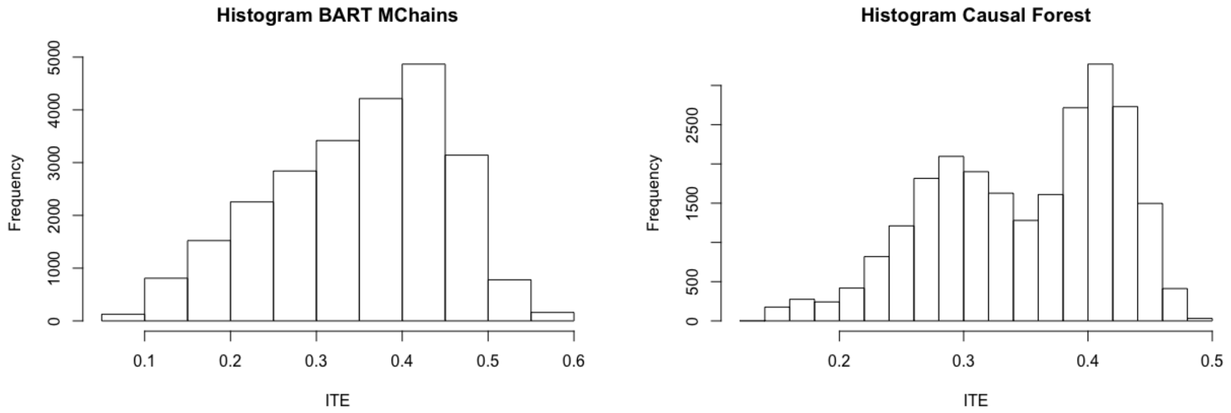
Table 14: Best Linear Predictor (BLP) heterogeneity test GSS

Notes: This table represents the Best Linear Predictor test, see Equation 15. In parenthesis, we denote the p-values for the hypothesis that the parameter β_1 is equal to zero. We can reject the null of no heterogeneity, if the coefficient is significantly greater than zero. The Generic framework is implemented with 100 splits.

method	BART MC	CF	GE BART	GE Boosting	GE Forest	GE Lasso	GE Nnet	GE Trees
Most	0.48	0.46	0.47	0.47	0.41	0.44	0.44	0.39
CI	(0.47, 0.50)	(0.43, 0.48)	(0.44,0.50)	(0.44,0.50)	(0.38,0.44)	(0.41,0.47)	(0.41,0.47)	(0.36,0.43)
Least	0.19	0.22	0.19	0.19	0.22	0.21	0.22	0.29
CI	(0.17, 0.20)	(0.20, 0.24)	(0.16,0.22)	(0.16,0.22)	(0.19,0.25)	(0.18,0.24)	(0.19,0.25)	(0.26,0.32)
Difference			0.28	0.28	0.19	0.23	0.22	0.10
CI			(0.24,0.33)	(0.24,0.33)	(0.14,0.22)	(0.19,0.28)	(0.18,0.27)	(0.07,0.16)

Table 15: Results of the Most and Least affected GATE estimation on the GSS

Notes: The 'Most' GATE is defined by the top 20% quantile of the treatment effect distribution. The 'Least' GATE is defined by the bottom 20% quantile of the treatment effect distribution. In the Generic method, we test for the difference between the Most and Least Affected Group, see Equation 17. The Generic framework is implemented with 100 splits.



(a) BART MChains

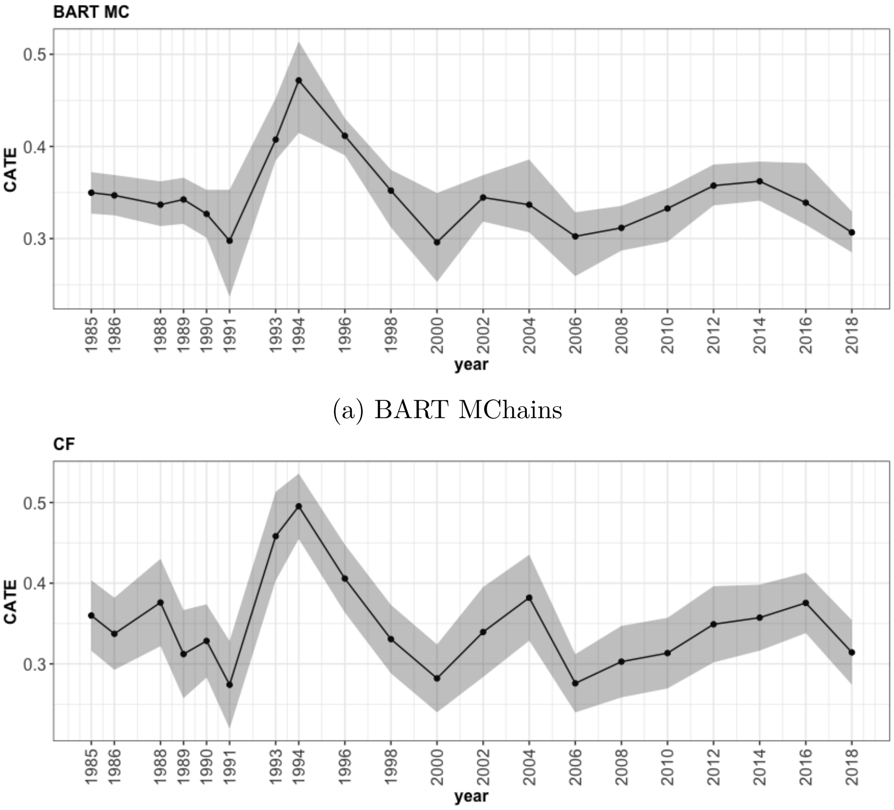
(b) Causal Forest

Figure 29: Histograms of the distribution of the Individual Treatment Effects (ITE) estimates of all respondents

Question: "On the average, Blacks have worse jobs, income, and housing than white people. Do you think these differences are ..." (*The General Social Survey*, n.d.)	Answer	BART MChains CATE	CI	Causal Forest CATE	CI
1.) Mainly due to discrimination?	Yes	0.27	(0.26, 0.29)	0.27	(0.25, 0.28)
	No	0.4	(0.39, 0.41)	0.4	(0.39, 0.42)
2.) Because most Blacks have less in-born ability to learn?	Yes	0.37	(0.35, 0.38)	0.37	(0.34, 0.4)
	No	0.35	(0.34, 0.36)	0.35	(0.34, 0.36)
3.) Because most Blacks don't have the chance for education that it takes to rise out of poverty?	Yes	0.3	(0.29, 0.31)	0.3	(0.29, 0.31)
	No	0.39	(0.38, 0.4)	0.4	(0.38, 0.41)
4.) Because most Blacks just don't have the motivation or will power to pull themselves up out of poverty?	Yes	0.39	(0.38, 0.4)	0.4	(0.38, 0.41)
	No	0.3	(0.29, 0.32)	0.3	(0.29, 0.31)

Table 16: CATEs of questions indicating negative attitude towards Blacks

Notes: For illustration purposes, BART MChains estimates a CATE of 0.27 for everyone who answers "Yes" to the question whether inequality is due to discrimination.

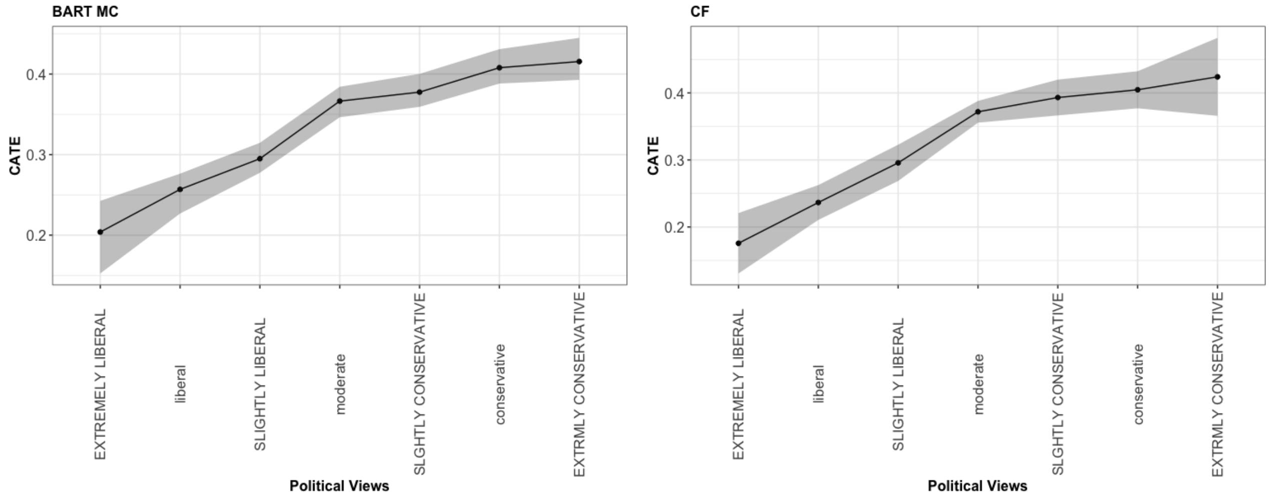


(a) BART MChains

(b) Causal Forest

Figure 30: Estimates of the Conditional Average Treatment Effects (CATE) for each year

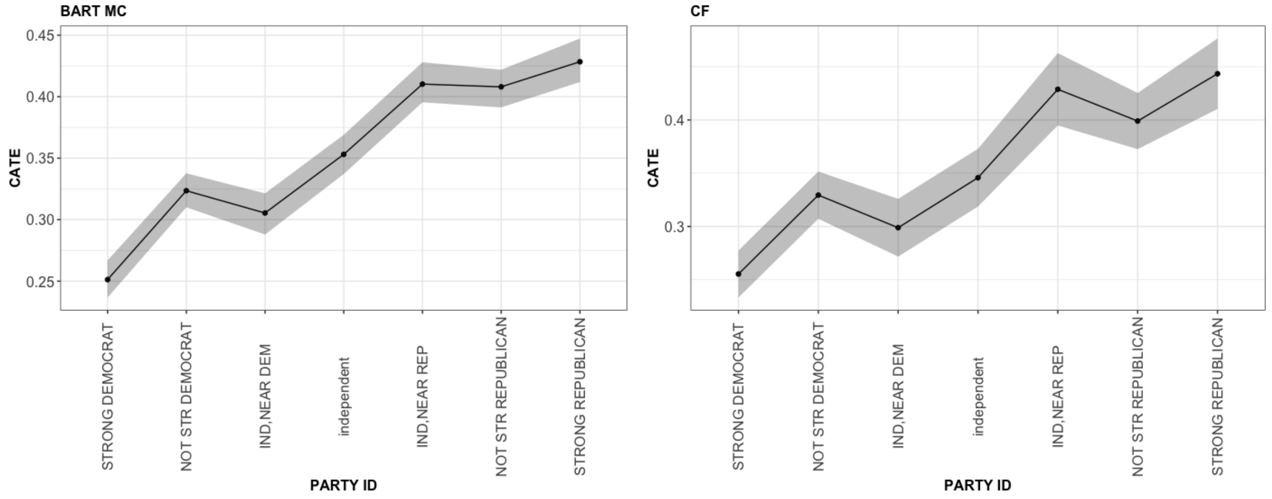
Note: The dots represent the CATE point estimates and the grey area the corresponding 95% confidence interval.



(a) BART MChains (b) Causal Forest

Figure 31: Estimates of CATEs for each political view

Note: The dots represent the CATE point estimates and the grey area the corresponding 95% confidence interval.



(a) BART MChains (b) Causal Forest

Figure 32: Estimates of CATEs for party membership

Note: The dots represent the CATE point estimates and the grey area the corresponding 95% confidence interval.

A.6 ML methods

In this section, we briefly describe the ML methods used in this research, including relevant tuning parameters.

A.6.1 Shrinkage methods

Shrinkage methods fit a model containing all p predictors, using a technique that regularizes less relevant estimates towards zero by introducing a penalty term (James, Witten, Hastie, & Tibshirani, 2013). Given the Residual Sum of Squares of Ordinary Least Squares denoted by $RSS = \sum_{i=1}^{n}\big(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\big)^2$, we minimize the Lasso loss function $L_{Lasso} = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$ or the Ridge loss function $L_{Ridge} = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$, where λ denotes a tuning parameter indicating the strength of the penalty term. Notably, the Lasso penalty term allows to shrink coefficients completely to zero, and thus can be considered as a variable selection method. In contrast, the Ridge penalty term can regularize coefficients close to zero, but coefficients cannot be excluded. Additionally, the Elastic Net is a combination of Lasso and Ridge, minimizing $L_{ElasticNet} = \frac{RSS}{2n} + \lambda(\frac{1-\alpha}{2} \sum_{j=1}^{p} \beta_j^2 + \frac{\alpha}{2} \lambda \sum_{j=1}^{p} |\beta_j|)$, which simplifies to the Lasso setting when $\alpha = 1$ and Ridge when $\alpha = 0$. Shrinkage methods have the advantage to dynamically adjust for the bias-variance trade-off: If we increase the tuning parameter λ, we shrink more variables in the model fit, which increases the bias, but decreases the variance (if we decrease λ, we have the vice-versa behaviour). We choose λ by minimizing a cross-validation error with ten-folds. In comparison with the other ML methods, shrinkage methods have the disadvantage that they require to manually specify non-linearities in the model specification. Since we are primarily interested in selecting the true confounders impacting the outcome and the treatment assignment, we focus on Lasso with $\alpha = 1$. To implement regularization methods, we use the *glmnet* R-package.

A.6.2 Tree

Tree-based methods divide the covariate space into simple subsets, whereby each subset is assigned to a predictive value of the outcome, e.g. the mean or the mode. Within the algorithm, we partition the covariate space by evaluating different decision rules. In regression settings, a suitable decision rule would be to minimize the Residual Sum of Squares, while in classification problems, to minimize an error rate such as entropy. Since it is computationally infeasible to consider all possible partitions within the data, trees follow a greedy approach, also known as recursive binary splitting. We start with an initial parent node P_0 including all observations and we split it into two children C_1 and C_2, while considering what is best at this particular step (James et al., 2013). Then, we continue splitting until each node has fewer observations than a predefined threshold, defining the terminal nodes. Setting a high threshold yields smaller trees, whereas a low threshold results in larger trees. Using the default parameters in the *rpart* R-package, we set the minimum observation required in a node to consider a split to 20, the minimum number of observations in a terminal node to seven and the maximum depth of any final node to 30. However, trees can become quite complex and can overfit, leading to low performance on a test data. Therefore, we apply cost-complexity pruning to build less complex subtrees T_s of the main tree T_0, which have a lower test error rate (James et al., 2013). Hereby, we determine a nested sequence of subtrees T_s, by recursively removing the least important splits based on a non-negative tuning parameter α, which trades off between the trees' complexity and the goodness of fit to the data. Formally, according to Friedman et al. (2001), for each value of α, there exist a subtree $T_s \subset T_0$, such that we minimize the complexity criterion $C_\alpha(T) = \sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T_s|$, where $|T|$ indicates the number of terminal nodes, R_m is the set belonging to the m^{th} terminal node and

$\hat{y}_{R_m}$ is the predictive outcome in this set. We choose the tuning parameter α using a ten-fold-cross-validation. Due to the dynamic splitting rules, trees are an ideal candidate to capture non-linearities and interactions.

A.6.3 Boosting

Boosting is an ensemble method which sequentially combines B decision trees, minimizing a loss function to boost their collective performance. Hereby, we consider the Gaussian (squared error) loss function in regression settings and the AdaBoost exponential loss function in classification problems (Friedman et al., 2001). The core idea of the Boosting approach is to take the residuals from each tree as input for the next tree in the sequence. Thereby, we slowly improve the tree in areas where performance is weak. There are several implementations of the Boosting approach and we use the gradient boosting algorithm implemented in the *gbm* R-package. Firstly, we initialize a tree t_0 by minimizing the corresponding loss function, formally $\hat{t}(x) = \text{argmin}_\rho \sum_{i=1}^{N} L(y_i, \rho)$. Then, for each tree in $b = 1, ..., B$, we calculate pseudo residuals by taking the negative gradient $r_i = -\frac{\partial}{\partial t(x_i)} L(y_i, t(x_i))$ evaluated at the tree from the previous step $t(x_i) = \hat{t}_{b-1}(x_i)$ (Friedman et al., 2001). Given these pseudo-residuals as dependent variable, we take a bagging fraction p of randomly selected observations and build regression tree with K splits. By varying K, we can account for the complexity for each tree. It can be beneficial to choose small K to focus the learning on specific covariates, which have challenging splitting criteria. Then, according to Ridgeway (2007), we compute the optimal node for each terminal node $k = 1, ..., K$ by $\rho_k = \text{argmin}_\rho \sum_{x_i \in R_k} L(y_i, \hat{t}_{b-1}(x_i) + \rho)$, where R_k is the region belonging to terminal node k. Lastly, we store the current tree by $\hat{t}_b(x) = \hat{t}_{b-1}(x) + \lambda \rho_{k(x)}$, where λ is a shrinkage parameter, which controls at which rate the boosting learns. In the training process, we consider 1000 trees/iterations, two splits per tree, a bagging fraction of 0.5 and a shrinkage parameter of 0.01. Furthermore, when predicting, we choose the best boosting fit by two-fold cross validation.

A.6.4 Random Forest

The Random Forest by Breiman (2001) is a large composition of de-correlated trees with the goal to combat over-fitting and to reduce prediction errors on a test data by lowering the variance. Hereby, for each tree in $b = 1, ..., B$, we take a bootstrap sample of the data and we build a tree T_b via recursive binary splitting, as explained in Section A.6.2. To de-correlate the trees, we randomly select a subset of m variables from all p predictor variables, where $m \leq p$. After iterating through all trees, we compute the final estimate by taking the averages. In a regression setting, we calculate the final estimates by $\hat{f}_{rf}^{B} = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$, whereas in a classification, we consider a pre-defined majority vote over all class prediction in each tree (Friedman et al., 2001).

Similar to the shrinkage methods, we can adjust for the bias-variance trade-off: If we decrease m, we decrease the variance at the cost of increasing the bias. Furthermore, each tree size is defined by the minimum size of terminal nodes, see Section A.6.2. Following the recommendations by Breiman (2001), we choose a subset of the predictor variables by $m = \frac{p}{3}$ and a minimum node size of $n = 5$ in a regression, while we choose $m = \sqrt{p}$ and $n = 1$ in classification. Moreover, we choose 250 trees within the forest, which is reasonable to receive stabilized results. Naturally, a Random Forest has the potential to outperform over-fitted single trees. In general, the performance of the Random Forest is comparable to the one of Boosting, but they are simpler to train and tune (Friedman et al., 2001). However, according to Friedman et al. (2001), the Random Forest performs worse than the Boosting when the amount of relevant variables is rather small in comparison to the amount of noisy predictors (e.g. two relevant variables and 50 noisy ones), since these relevant variables might be missed when taking the random subsets for the splits of each tree.

A.6.5 Neural Net

A Neural Network can be considered as a two-stage nonlinear generalization of a linear model, both for regression and classification (Friedman et al., 2001). The network consists of an input layer, a hidden layer and an output layer. In the first stage, a weighted linear combination of the input covariates matrix X is transformed via a nonlinear activation function $\sigma(v)$, to calculate features in the hidden layer. Formally, for $1, ..., M$ derived features in the hidden layer, we define $Z_m = \sigma(\alpha_{0m} + \alpha_m^T X)$, where $Z = (Z_1, ..., Z_M)$ are the derived features, α_{0m} is a bias capturing the intercept and the vector α_m denotes the covariate weights. A frequently used activation function is the sigmoid function $\sigma(v) = \frac{1}{1+e^{-v}}$ (Friedman et al., 2001). In the second stage, a weighted linear combination of features from the hidden layer Z_m is used to estimate the output variable. In a regression setting, we have $Y = g(\beta_0 + \beta^T Z)$, where $g(\cdot)$ is a function specifying the final transformation of the hidden units, β_0 is a bias capturing the intercept and β denotes a weight vector. In a classification setting, we would consider the different levels of the output layer.

We solve for the unknown weight parameters via back-propagation and gradient descent (Friedman et al., 2001). Back-propagation is a technique used to derive the relevant gradients over chain rules from the last layer to the first layer, while gradient decent is a method to minimize the resulting loss functions by ‘walking down’ the derivative in a negative direction. When training Neural Nets, we have to consider that they can overfit when too many weights are considered. Therefore, a *decay* parameter is introduced, which reflects a regularization term in the loss functions of the weights optimization (Friedman et al., 2001). If we increase the *decay* parameter, we shrink some weights to zero, thereby increasing the bias on the training data fit, but improving the performance on test data due to lower variance. Following standard recommendations by Friedman et al. (2001), we choose a *decay* parameter of 0.01. Furthermore, the authors recommend to standardize all inputs to have mean zero and standard deviation one, so that representative initial values are chosen when minimizing the loss functions and all weights are treated equally in the regularization process. Lastly, we choose two hidden layers and use the R-package *nnet*.

A.6.6 Tuning parameters of ML methods

ML method	**parameters**
Lasso (glmnet)	$\alpha = 1$; λ is chosen by the lowest 10 fold cross-validation error
Trees (rpart)	method: "anova" for regression and "class" for classification; minimum number of observations to consider at each split = 20*; minimum size of terminal nodes = 7*; maximum depth of any node of the final tree = 30*; complexity parameter (cp) is chosen by 10-fold cross validation
Boosting (gbm)	number of trees/iterations for fitting the model = 1000; number of trees/iterations for prediction is chosen by 2-fold cross-validation; distribution: Gaussian for regression and Adaboost for classification; interaction depth of each tree = 2; minimum size of terminal nodes = 1; shrinkage = 0.01*; bag fraction = 0.5*;
Random Forest (randomForest)	number of trees = 250; minimum size of terminal nodes: 5 for regression and 1 for classification*; Number of variables randomly sampled as candidates at each split: $\frac{p}{3}$ for regression and $\sqrt{p}$ for classification (p is the number of control variables in the data set)*
Neural Net (nnet)	number of units in the hidden layer = 2; parameter for weight decay = 0.01

Table 17: Overview Tuning parameters ML methods

Notes: This table summarizes the input parameters used for the Machine Learning methods. The name in brackets behind the ML methods denotes the corresponding R-package. Parameters with an asterisk* denote the default parameter given by the R-package. For a detailed description of the parameters see Appendix A.6.1 - A.6.5.

A.7 Sensitivity check ML input parameters

In the following, we conduct a sensitivity check of the input parameters (see Appendix A.6.6 Table 17) of Lasso, Tree, Boosting, Random Forest and Neural Net used within the DML and the Generic framework. To accomplish that, we tune the parameters over a grid, using the R-package *caret* (Kuhn et al., 2008). We omit the BART-based methods because they are rather insensitive to the choice of tuning parameters (Chipman et al., 2010; Green & Kern, 2012) and we omit the Causal Forest as its parameters are already tuned, see Section 3.4.3.

Tuning all parameters within the DML and the Generic framework is computationally challenging due to the dimension of the simulation settings in combination with the additional sample splitting. Firstly, we need to tune the parameters for the three nuisance functions separately ($E[Y|D = 1X], E[Y|D = 0, X]$ and $E[D|X]$). Secondly, for each knob we need to tune over all simulation replications (100 data sets per knob). Lastly, for each single simulation replication, we need to tune each sample splitting step within the DML and the Generic framework (10 splits in this research). We omit the last tuning process at the sample splitting step, since it is too computationally intensive. However, we still consider the first two dimensions, leading to 300 tuning processes per knob.

For each method, we consider the available tuning parameter combinations shown in Tables 18 - 22. The parameters not listed in the grid are equal to the ones in Table 17. For each candidate tuning parameter combination, we fit a model on resampled training data via repeated 2-fold cross-validation. Next, we choose the best tuning parameter combination by the lowest aggregated RMSE over the hold-out sample sets, see Kuhn et al. (2008) for more details. The grid is randomly chosen for the shrinkage methods and for the Tree, whereas we manually define the grid for Boosting, Random Forest and Neural Net. Lastly, the best tuning parameters are used as inputs for the ML methods in the DML and Generic framework at each simulation replication step. Given these tuned input parameters, we revisit the results of the Benchmark Analysis (Table 4, 7, 10 and 11).

#grid	α	λ	#grid	α	λ	#grid	α	λ	#grid	α	λ	#grid	α	λ
1	0.1	0.001	21	0.3	0.001	41	0.5	0.001	61	0.7	0.001	81	0.9	0.001
2	0.1	0.003	22	0.3	0.003	42	0.5	0.003	62	0.7	0.003	82	0.9	0.003
3	0.1	0.008	23	0.3	0.008	43	0.5	0.008	63	0.7	0.008	83	0.9	0.008
4	0.1	0.018	24	0.3	0.018	44	0.5	0.018	64	0.7	0.018	84	0.9	0.018
5	0.1	0.042	25	0.3	0.042	45	0.5	0.042	65	0.7	0.042	85	0.9	0.042
6	0.1	0.096	26	0.3	0.096	46	0.5	0.096	66	0.7	0.096	86	0.9	0.096
7	0.1	0.223	27	0.3	0.223	47	0.5	0.223	67	0.7	0.223	87	0.9	0.223
8	0.1	0.514	28	0.3	0.514	48	0.5	0.514	68	0.7	0.514	88	0.9	0.514
9	0.1	1.188	29	0.3	1.188	49	0.5	1.188	69	0.7	1.188	89	0.9	1.188
10	0.1	2.744	30	0.3	2.744	50	0.5	2.744	70	0.7	2.744	90	0.9	2.744
11	0.2	0.001	31	0.4	0.001	51	0.6	0.001	71	0.8	0.001	91	1.0	0.001
12	0.2	0.003	32	0.4	0.003	52	0.6	0.003	72	0.8	0.003	92	1.0	0.003
13	0.2	0.008	33	0.4	0.008	53	0.6	0.008	73	0.8	0.008	93	1.0	0.008
14	0.2	0.018	34	0.4	0.018	54	0.6	0.018	74	0.8	0.018	84	1.0	0.018
15	0.2	0.042	35	0.4	0.042	55	0.6	0.042	75	0.8	0.042	95	1.0	0.042
16	0.2	0.096	36	0.4	0.096	56	0.6	0.096	76	0.8	0.096	96	1.0	0.096
17	0.2	0.223	37	0.4	0.223	57	0.6	0.223	77	0.8	0.223	97	1.0	0.223
18	0.2	0.514	38	0.4	0.514	58	0.6	0.514	78	0.8	0.514	98	1.0	0.514
19	0.2	1.188	39	0.4	1.188	59	0.6	1.188	79	0.8	1.188	99	1.0	1.188
20	0.2	2.744	40	0.4	2.744	60	0.6	2.744	80	0.8	2.744	100	1.0	2.744

Table 18: Tuning parameter Shrinkage method

Notes: Each row in a block represent a possible combination of α and λ as inputs for the penalty term of the Shrinkage method. Lasso is given by $\alpha = 1$ and Elastic Net by $0 < \alpha < 1$ (Friedman et al., 2001). The grid is randomly chosen at each tuning iteration.

#grid	1	2	3	4	5	6	7	8	9	10
complexity parameter	0.009	0.011	0.011	0.019	0.032	0.042	0.051	0.062	0.172	0.263

Table 19: Tuning parameter Tree grid

Notes: Each value represents a possible complexity parameter as input for the Tree method. The grid is randomly chosen at each tuning iteration.

#grid	1	2	3	4	5	6	7	8	9	10	11	12
number trees	600	600	600	1000	1000	1000	600	600	600	1000	1000	1000
interaction depth	2	3	4	2	3	4	2	3	4	2	3	4
min. observations terminal node	1	1	1	1	1	1	5	5	5	5	5	5

Table 20: Tuning parameter Boosting grid

Notes: Each column represents a possible combination of the numbers of trees, the interaction depth and the minimum observations in the terminal node as input for the Boosting method. The grid is manually predefined.

#grid	1	2	3
number trees	500	500	500
numbers of variables randomly sampled in regressions	22	27	32

Table 21: Tuning parameter Random Forest grid

Notes: Each column represents a possible combination of the numbers of trees and the numbers of variables randomly sampled in regressions as input for the Random Forest method. The grid is manually predefined.

#grid	1	2	3	4	5	6
size	2	4	8	2	4	8
decay	0.01	0.01	0.01	0.02	0.02	0.02

Table 22: Tuning parameter Neural Net grid

Notes: Each column represents a possible combination of the size and decay parameter as inputs for the Neural Net method. The grid is manually predefined.

knob	BART MC	Causal Forest	DML BART	DML Boost*	DML Lasso*	DML Nnet*	DML RF*	DML Trees*	GE BART	GE Boost*	GE Lasso*	GE Nnet	GE RF*	GE Trees*	OLS
27	0.01 (0.97)	0.02 (0.87)	0.01 (0.96)	0.01 (0.96)	0.03 (0.81)	0.03 (0.83)	0.02 (0.90)	0.02 (0.89)	0.01 (0.96)	0.01 (0.97)	0.03 (0.89)	0.04 (0.74)	0.02 (0.85)	0.02 (0.87)	0.07 (0.67)
55	0.01 (0.97)	0.03 (0.79)	0.01 (0.98)	0.01 (0.96)	0.05 (0.72)	0.04 (0.85)	0.01 (0.97)	0.02 (0.94)	0.01 (0.98)	0.01 (0.98)	0.05 (0.77)	0.05 (0.66)	0.02 (0.76)	0.02 (0.97)	0.08 (0.69)
40	0.01 (0.98)	0.03 (0.80)	0.01 (0.96)	0.01 (0.96)	0.04 (0.79)	0.03 (0.84)	0.02 (0.87)	0.02 (0.89)	0.01 (0.95)	0.01 (0.96)	0.04 (0.87)	0.04 (0.63)	0.02 (0.86)	0.02 (0.91)	0.06 (0.63)
21	0.01 (0.92)	0.03 (0.68)	0.03 (0.78)	0.03 (0.75)	0.05 (0.72)	0.04 (0.87)	0.03 (0.74)	0.03 (0.82)	0.03 (0.85)	0.03 (0.76)	0.04 (0.80)	0.05 (0.65)	0.03 (0.86)	0.03 (0.84)	0.05 (0.64)
31	0.01 (0.92)	0.03 (0.83)	0.01 (0.99)	0.01 (0.99)	0.03 (0.80)	0.03 (0.91)	0.02 (0.96)	0.02 (0.90)	0.01 (0.92)	0.01 (0.93)	0.03 (0.80)	0.04 (0.56)	0.02 (0.86)	0.02 (0.95)	0.06 (0.65)
29	0.01 (0.98)	0.05 (0.49)	0.01 (0.90)	0.01 (0.82)	0.05 (0.52)	0.06 (0.59)	0.03 (0.65)	0.04 (0.56)	0.01 (0.94)	0.01 (0.92)	0.05 (0.60)	0.04 (0.62)	0.02 (0.90)	0.04 (0.75)	0.11 (0.31)
28	0.01 (0.92)	0.04 (0.59)	0.02 (0.89)	0.02 (0.93)	0.06 (0.55)	0.06 (0.54)	0.03 (0.71)	0.03 (0.72)	0.01 (0.91)	0.01 (0.91)	0.06 (0.57)	0.05 (0.54)	0.02 (0.83)	0.03 (0.88)	0.12 (0.16)

Table 23: Sensitivity check Benchmarking Analysis results ATE: RMSE and (Coverage)

Notes: This table shows the results of the sensitivity check of the ML input parameters compared to the main results ATE Benchmarking Analysis in Table 4 (in terms of RMSE and coverage in parenthesis). The input parameters of all methods with an asterisk* have been chosen by the best combinations resulting from the tuning grids defined in Appendix Tables 18-22. The best combination has been chosen separately for each nuisance function (E[Y|D=1,X], E[Y|D=0,X] and E[D|X]) and for each simulation replication per knob (r=100).

knob	BART MC	Causal Forest	DML BART	DML Boost*	DML Lasso*	DML Nnet*	DML RF*	DML Trees*	GE BART	GE Boost*	GE Lasso*	GE Nnet*	GE RF*	GE Trees*	OLS
27	-0.00	-0.01	-0.00	-0.00	-0.01	-0.01	-0.01	-0.01	0.00	-0.00	-0.01	0.02	0.01	-0.01	-0.02
	(0.03)	(0.06)	(0.05)	(0.05)	(0.08)	(0.10)	(0.05)	(0.06)	(0.05)	(0.04)	(0.09)	(0.09)	(0.05)	(0.07)	(0.08)
55	-0.00	-0.00	-0.00	-0.00	-0.01	-0.01	-0.00	-0.00	0.01	-0.00	-0.01	0.02	0.01	0.00	-0.01
	(0.03)	(0.06)	(0.05)	(0.05)	(0.07)	(0.09)	(0.05)	(0.06)	(0.04)	(0.04)	(0.09)	(0.08)	(0.05)	(0.06)	(0.08)
40	-0.00	-0.01	-0.00	-0.00	-0.01	-0.01	-0.00	-0.00	0.00	-0.00	-0.01	0.03	0.01	-0.00	-0.01
	(0.03)	(0.06)	(0.05)	(0.05)	(0.07)	(0.09)	(0.05)	(0.06)	(0.04)	(0.04)	(0.09)	(0.08)	(0.05)	(0.07)	(0.08)
21	0.00	-0.01	-0.00	-0.00	-0.01	-0.01	-0.00	-0.01	-0.00	-0.01	-0.01	0.03	0.00	-0.00	-0.01
	(0.04)	(0.06)	(0.05)	(0.05)	(0.08)	(0.12)	(0.05)	(0.07)	(0.06)	(0.06)	(0.10)	(0.10)	(0.07)	(0.08)	(0.09)
31	0.00	-0.01	-0.00	-0.00	-0.01	-0.01	-0.00	-0.01	0.01	0.00	-0.01	0.03	0.01	-0.00	-0.01
	(0.03)	(0.07)	(0.06)	(0.06)	(0.07)	(0.10)	(0.07)	(0.07)	(0.05)	(0.05)	(0.07)	(0.08)	(0.06)	(0.08)	(0.08)
29	-0.00	-0.01	-0.01	-0.01	-0.03	-0.03	-0.02	-0.03	0.00	-0.00	-0.02	0.01	0.00	-0.02	-0.05
	(0.03)	(0.06)	(0.05)	(0.04)	(0.07)	(0.09)	(0.06)	(0.07)	(0.05)	(0.04)	(0.07)	(0.07)	(0.06)	(0.07)	(0.08)
28	-0.00	-0.00	-0.01	-0.01	-0.03	-0.03	-0.01	-0.02	-0.00	-0.01	-0.03	0.01	0.01	-0.01	-0.05
	(0.03)	(0.07)	(0.06)	(0.06)	(0.08)	(0.10)	(0.07)	(0.07)	(0.05)	(0.04)	(0.08)	(0.08)	(0.06)	(0.07)	(0.09)

Table 24: Sensitivity check Benchmarking Analysis results ATE: Bias and (Int Length)

Notes: This table shows the results of the sensitivity check of the ML input parameters compared to the main results ATE Benchmarking Analysis in Table 10 (in terms of bias and interval length in parenthesis). The input parameters of all methods with an asterisk* have been chosen by the best combinations resulting from the tuning grids defined in Appendix Tables 18-22. The best combination has been chosen separately for each nuisance function (E[Y|D=1,X], E[Y|D=0,X] and E[D|X]) and for each simulation replication per knob (r=100)

knob	BART MC		Causal Forest		GE BART		GE Boost*		GE Lasso*		GE Nnet*		GE RF*		GE Trees*	
	most	least	most	least	most	least	most	least	most	least	most	least	most	least	most	least
28	0.05	0.06	0.10	0.10	0.10	0.08	0.06	0.07	0.23	0.20	0.28	0.24	0.11	0.10	0.20	0.21
	(0.66)	(0.64)	(0.54)	(0.37)	(0.57)	(0.59)	(0.84)	(0.76)	(0.48)	(0.39)	(0.14)	(0.16)	(0.48)	(0.51)	(0.22)	(0.14)
56	0.06	0.06	0.09	0.09	0.10	0.09	0.07	0.07	0.24	0.32	0.28	0.29	0.10	0.08	0.16	0.17
	(0.70)	(0.65)	(0.50)	(0.52)	(0.62)	(0.64)	(0.78)	(0.82)	(0.36)	(0.30)	(0.20)	(0.24)	(0.30)	(0.54)	(0.31)	(0.28)
41	0.06	0.05	0.08	0.11	0.09	0.09	0.07	0.07	0.19	0.22	0.26	0.26	0.11	0.10	0.20	0.21
	(0.70)	(0.63)	(0.62)	(0.39)	(0.58)	(0.65)	(0.75)	(0.74)	(0.51)	(0.36)	(0.14)	(0.13)	(0.43)	(0.52)	(0.21)	(0.16)
22	0.10	0.07	0.12	0.12	0.15	0.12	0.14	0.10	0.29	0.64	0.31	0.26	0.17	0.12	0.28	0.27
	(0.66)	(0.62)	(0.70)	(0.51)	(0.62)	(0.67)	(0.63)	(0.76)	(0.45)	(0.36)	(0.21)	(0.26)	(0.66)	(0.75)	(0.30)	(0.26)
32	0.06	0.05	0.13	0.15	0.10	0.09	0.06	0.08	0.20	0.22	0.30	0.26	0.13	0.13	0.25	0.26
	(0.45)	(0.45)	(0.43)	(0.28)	(0.52)	(0.66)	(0.77)	(0.73)	(0.46)	(0.35)	(0.12)	(0.12)	(0.35)	(0.42)	(0.18)	(0.09)
30	0.06	0.06	0.11	0.12	0.11	0.10	0.10	0.08	0.20	0.24	0.28	0.25	0.13	0.12	0.21	0.24
	(0.67)	(0.54)	(0.69)	(0.20)	(0.61)	(0.66)	(0.74)	(0.69)	(0.55)	(0.29)	(0.22)	(0.12)	(0.40)	(0.35)	(0.23)	(0.08)
27	0.04	0.04	0.06	0.06	0.06	0.07	0.04	0.05	0.19	0.19	0.24	0.20	0.09	0.07	0.16	0.17
	(0.73)	(0.74)	(0.71)	(0.58)	(0.60)	(0.68)	(0.72)	(0.71)	(0.45)	(0.52)	(0.13)	(0.26)	(0.40)	(0.64)	(0.32)	(0.17)

Table 25: Sensitivity check Benchmarking Analysis results GATE: RMSE and (Coverage)

Notes: This table shows the results of the sensitivity check of the ML input parameters compared to the main results GATE Benchmarking Analysis in Table 7 (in terms of RMSE and coverage in parenthesis). The input parameters of all methods with an asterisk* have been chosen by the best combinations resulting from the tuning grids defined in Appendix Tables 18-22. The best combination has been chosen separately for each nuisance function (E[Y|D=1,X], E[Y|D=0,X] and E[D|X]) and for each simulation replication per knob (r=100).

knob	BART MC		Causal Forest		GE BART		GE Boost*		GE Lasso*		GE Nnet*		GE RF*		GE Trees*	
	most	least	most	least	most	least	most	least	most	least	most	least	most	least	most	least
28	0.04	-0.04	0.05	-0.08	0.05	-0.05	0.03	-0.04	0.10	-0.15	0.22	-0.20	0.09	-0.08	0.17	-0.18
	(0.09)	(0.09)	(0.13)	(0.13)	(0.14)	(0.14)	(0.12)	(0.12)	(0.19)	(0.19)	(0.20)	(0.19)	(0.15)	(0.14)	(0.17)	(0.17)
56	0.04	-0.04	0.06	-0.06	0.05	-0.04	0.03	-0.03	0.10	-0.21	0.21	-0.22	0.09	-0.06	0.13	-0.13
	(0.10)	(0.10)	(0.12)	(0.12)	(0.12)	(0.12)	(0.11)	(0.11)	(0.23)	(0.23)	(0.21)	(0.21)	(0.13)	(0.13)	(0.15)	(0.15)
41	0.04	-0.04	0.05	-0.07	0.05	-0.05	0.04	-0.04	0.11	-0.15	0.22	-0.20	0.09	-0.07	0.17	-0.18
	(0.10)	(0.10)	(0.14)	(0.12)	(0.14)	(0.14)	(0.12)	(0.12)	(0.19)	(0.19)	(0.19	(0.19	(0.14)	(0.14)	(0.18)	(0.18)
22	0.05	-0.05	0.05	-0.08	0.03	-0.05	0.01	-0.03	0.07	-0.23	0.26	-0.20	0.08	-0.08	0.21	-0.21
	(0.13)	(0.12)	(0.17)	(0.16)	(0.21)	(0.20)	(0.18)	(0.18)	(0.27)	(0.27)	(0.25)	(0.24)	(0.27)	(0.25)	(0.21)	(0.21)
32	0.05	-0.05	0.09	-0.12	0.06	-0.05	0.04	-0.05	0.12	-0.16	0.26	-0.23	0.11	-0.11	0.22	-0.24
	(0.09)	(0.08)	(0.16)	(0.14)	(0.15)	(0.15)	(0.14)	(0.14)	(0.19)	(0.19)	(0.20)	(0.20)	(0.16)	(0.16)	(0.20)	(0.20)
30	0.04	-0.05	0.04	-0.10	0.05	-0.06	0.04	-0.05	0.08	-0.18	0.20	-0.21	0.09	-0.10	0.17	-0.22
	(0.09)	(0.08)	(0.14)	(0.12)	(0.14)	(0.14)	(0.12)	(0.12)	(0.18)	(0.18)	(0.19)	(0.18)	(0.15)	(0.15)	(0.18)	(0.18)
27	0.03	-0.03	0.03	-0.05	0.04	-0.03	0.03	-0.04	0.13	-0.13	0.20	-0.16	0.08	-0.06	0.13	-0.15
	(0.09)	(0.08)	(0.12)	(0.12)	(0.12)	(0.12)	(0.11)	(0.11)	(0.21)	(0.21)	(0.20)	(0.19)	(0.13)	(0.13)	(0.16)	(0.16)

Table 26: Sensitivity check Benchmarking Analysis results GATE: Bias and (Int Length)

Notes: This table shows the results of the sensitivity check of the ML input parameters compared to the main results GATE Benchmarking Analysis in Table 11 (in terms of bias and interval length in parenthesis). The input parameters of all methods with an asterisk* have been chosen by the best combinations resulting from the tuning grids defined in Appendix Tables 18-22. The best combination has been chosen separately for each nuisance function (E[Y|D=1,X], E[Y|D=0,X] and E[D|X]) and for each simulation replication per knob (r=100).

A.8 Derivations

In the following, we solve the sample analog of the score function in the interactive model of the DML approach inspired by Robins and Rotnitzky (1995) for β_0, see Equation (12). Note that according to the DML-2 procedure, the nuisance parameters which are trained on the complementary sample via ML methods are pooled over K folds. Superscript $^{(-i)}$ denotes that the nuisance parameter are estimated via ML-methods on the complementary data sets.

$$\frac{1}{N}\sum_i \rho(W_i;\beta_0,\delta_0) = 0 \tag{36}$$

$$\Rightarrow \frac{1}{N}\sum_i [g_0(1,X_i) - g_0(0,X_i)) + \frac{D_i(Y_i - g_0(1,X_i)}{m_0(X_i)} - \frac{(1-D_i)(Y_i - g_0(0,X_i))}{1-m_0(X_i)} - \beta_0] = 0$$

$$\Rightarrow \frac{1}{N}\sum_i \beta_0 = \frac{1}{N}\sum_i [g_0(1,X_i) - g_0(0,X_i)) + \frac{D_i(Y_i - g_0(1,X_i)}{m_0(X_i)} - \frac{(1-D_i)(Y_i - g_0(0,X_i))}{1-m_0(X_i)}]$$

$$\Rightarrow \hat{\beta}_0 = \frac{1}{N}\sum_i [g_0^{(-i)}(1,X_i) - g_0^{(-i)}(0,X_i)) + \frac{D_i(Y_i - g_0^{(-i)}(1,X_i)}{m_0^{(-i)}(X_i)} - \frac{(1-D_i)(Y_i - g_0^{(-i)}(0,X_i))}{1-m_0^{(-i)}(X_i)}]$$

Hence, $\hat{\beta}_0$ is equal to the mean over all observations denoted by $\hat{\beta}_0 = \frac{1}{N}\sum_{i=1}^{N} {Y_i^*}_{DR}(W_i, \hat{\delta}_0^{(-i)})$.

A.9 Algorithms

Algorithm 4: Gradient tree by Athey et al. (2019)

```
function gradienttree(set of examples $J$, domain $X$)
Initialize parent node:   $P_0 \leftarrow createnode(J, X)$
Initialize a queue with a single element:   $Q \leftarrow initializequeue(P_0)$
while notnull(node P ← pop(Q)) do
    Compute Equation (21) to retrieve estimates $(\hat{\beta}_P, \hat{\delta}_P)$
    Compute Equation (24) to retrieve $\hat{A}_p$
    Use $(\hat{\beta}_P, \hat{\delta}_P, \hat{A}_p)$ to solve Equation (25) over $P$ to get vector of potential outcomes $R_p$
    Optimize Equation (26) to return split $\Sigma$
    if split $\Sigma$ is possible then
        Given node $P$ split into children $C_1$ and $C_2$
        Add parent node and children node to queue
    end if
end while
return tree with node $P_0$
```