TI 2021-090/III
Tinbergen Institute Discussion Paper

Finite Sample Evaluation of Causal Machine Learning Methods: Guidelines for the Applied Researcher

Andrea A. Naghi[1]
Christian P. Wirths[1]

[1] Erasmus University Rotterdam

Finite Sample Evaluation of Causal Machine Learning Methods: Guidelines for the Applied Researcher

Andrea A. Naghi[*] and Christian P. Wirths [†]

August 2021

Abstract: The econometrics literature proposed several new causal machine learning methods (CML) in the past few years. These methods harness the strength of machine learning methods to flexibly model the relationship between the treatment, outcome and confounders, while providing valid inferential statements. Whereas numerous options are available now to the applied economics researcher, there is limited guidance on the most useful methodology for a particular applied setting. In this paper, we perform a comprehensive evaluation of the finite sample performance of recently introduced CML methods from the econometrics literature, under a wide range of data generating processes. We focus our analysis on data features that are relevant for causal inference such as varying degrees of: nonlinearity in the outcome and treatment equations, overlap, percentage of treated, alignment and heterogeneity in the treatment effect. We evaluate the methods that have received the most attention so far from the empirical economics literature: double machine learning, causal forest and the generic machine learning methods. Results on the bias, root mean squared error, coverage rates and interval lengths for the average treatment effect, group average treatment effects and individual treatment effects reveal information on the characteristics of the methods and the data features that affect their performance the most.

Keywords: average treatment effect, causal inference, empirical Monte Carlo, heterogeneous treatment effects, individual treatment effects, machine learning,

J.E.L. Classification: C01, C21, D04

[*] Corresponding author. Department of Econometrics, Erasmus University and Tinbergen Institute. Email: naghi@ese.eur.nl. Naghi acknowledges partial support from EU Horizon 2020, Marie Skłodowska-Curie individual grant (No. 797286).

[†] Department of Econometrics, Erasmus University. Email: wirths@ese.eur.nl.

1 Introduction

The burgeoning econometrics literature on causal machine learning methods (CML) from the last few years provides some promising new tools for causal analysis. These techniques combine machine learning (ML) methods with causal inference questions, while establishing theoretical results on the consistency, asymptotic normality and validity of confidence intervals of the causal parameters of interest (see, e.g., Chernozhukov et al. 2018a, Wager and Athey 2018, Athey et al. 2019, Chernozhukov et al. 2018b; and some of the first empirical studies using these methods: Bertrand et al. 2017, Davis and Heller 2017, Deryugina et al. 2019, Knaus et al. 2017, Strittmatter 2019). Causal machine learning methods prove to be especially useful when the researcher needs to control for many covariates (raw or technical controls) and thus requires a more flexible approach to fit the model. Baiardi and Naghi (2021) highlight through revisited studies the added value of these new techniques relative to traditional causal inference methods. While a large number of ML methods for causal inference have recently become available in the econometrics literature, there is no comprehensive Monte Carlo study to evaluate and compare their performance. This naturally leads to harder decisions from the empirical researcher's perspective regarding the method to be employed and defended in a particular applied setting.

The purpose of this paper is to evaluate the finite sample performance of recently developed causal machine learning methods from the econometrics literature across a wide range of data features relevant for causal estimation, and thus to provide guidelines for the applied researcher on the use of these methods. A well suited vehicle for our aim is the "2016 Atlantic Causal Inference Conference Competition" initiated by Dorie et al. (2019), where authors of 30 different causal inference methods from the machine learning and statistics literature submitted their method for evaluation. This competition has several attractive features that we exploit. First, it provides a comprehensive framework to evaluate the causal effect in observational studies across a much broader set of data generating processes (DGPs) than what is commonly performed in a typical methodological paper that introduces a CML method, i.e., 77 different simulation scenarios. Second, the competition calibrates the simulations to real-life scenarios in the sense that it accounts for different covariate types encountered in practice (continuous, categorical, binary), with a joint distribution coming from real data, instead of generating them from a multivariate normal distribution, for example. To this end, the covariates are selected from a real study, the Collaborative Perinatal Project (Niswander and Gordon, 1972), a large panel data on women and their children, the aim of which is to examine the causal factors of different development disorders. Consequently, the simulation design can mimic natural associations that could arise between the covarites. On the other hand, the outcome and treatment variables are simulated, in order to be able to manipulate and measure the different empirically relevant data features on which the methods are evaluated. Third, the data generating processes are constructed such that they exhibit a range of data complications commonly encountered in practice and relevant for causal inference. Thus, they present a varying degree of: nonlinearity in the outcome and treatment equations, overlap, percentage of treated, alignment and heterogeneity of the treatment effect.

In this paper, we focus on recently developed causal machine learning methods from the econometrics literature, with well established theoretical properties, which have not been directly compared. We evaluate the following approaches: the Double/Debiased Machine Learning (DML) method (Chernozhukov et al. 2017, Chernozhukov et al. 2018a) combined with Lasso, Trees, Neural Net, Random Forest and Boosting; the Causal Forest (CF) (Wager and Athey (2018), Athey et al. 2019); the Generic Machine Learning Method (GML) (Chernozhukov et al. 2018b) combined with Lasso, Trees, Neural Net, Random Forest and Boosting; the Doubly Robust Modified Outcome Method (DR MOM)[1] (Knaus et al., 2018) combined with Lasso, Trees, Neural Net, Random Forest and Boosting. On top of these main methods, we are interested in also adding one of the best performing methods from the Atlantic Causal Inference Competition, the BART MChains (uses several chains with distinct starting points, see Dorie et al. 2019), to compare the performance of the newly developed econometric causal machine learning methods with this top performing method. To make comparisons with traditional econometric approaches used for causal inference, we also add the linear model with interactions, estimated by the ordinary least squares (OLS) to our pool of methods. Finally, we propose new CML methods that we add to our list of competitors by combining BART with the DML, with the GML and with the DR MOM methods.

In our evaluation, we are targeting the average treatment effect (ATE), as well as heterogeneous treatment effects in the form of group average treatment effects (GATEs)

[1] Note that theoretical properties are not established for the DR MOM method, but it is one of the few methods that allows the computation of individual treatment effects; thus we include it in our analysis.

and individual treatment effects (ITEs). Specifically, we estimate the ATE with DML, CF, GML, BART MChains and OLS; the GATEs with CF, GML and BART MChains; and the ITEs with CF, DR MOM and BART MChains. For a complete picture on the methods' performance, for the ATE and GATEs, we provide results on the bias, root mean squared error (RMSE), confidence interval coverage rate and confidence interval length. For the ITEs, we compute the precision in estimating heterogeneous effects (PEHE). Our analysis looks at both overall performance and performance by different criteria - the criteria being the data features mentioned above: nonlinearity in the outcome and treatment equations, overlap, percentage of treated, alignment and heterogeneity of the treatment effect. For the overall performance, the results of each method are aggregated over all 77 simulation scenarios. In the analysis by the different criteria, we study the impact of varying each of these data characteristics on the competing methods.

When looking at the overall performance across all 77 DGPs, our results indicate that the best performing method for both average treatment effect and heterogeneous treatment effects is BART MChains, followed by our newly introduced methods, DML BART and GML BART, and then by the Boosting and Random Forest combinations of DML, GML and DR MOM. Further, we conclude that when the overlap assumption is violated or treatment effect heterogeneity is higher, all methods perform worse, some of them being more affected than others. Increasing the percentage of treated observations does not lead to notable changes in the performance of the methods. Increasing the level of alignment[2] leads to a mixed performance, with the BART-based methods being relatively more robust. Step functions in the DGP, tend to be more suited for Tree-based CML methods, while Lasso-based CML methods perform worse in this case. Completely linear DGPs, negatively affect the performance of Tree-based methods. The OLS, one of the most used estimators (for causal inference) in economics is consistently outperformed by the causal machine learning methods under the different data complications under consideration. Sensitivity analyses show that our results are robust to tuning parameter choices in the causal machine learning methods.

While we base our simulation study on the setup of the Atlantic Causal Inference Competition, our paper differs from Dorie et al. (2019) in several aspects. First, we focus on newly developed causal machine learning methods from the econometrics lit-

[2] Alignment refers to the degree to which the treatment and outcome equations share the same confounding terms. See section 2.2 for more details.

erature, which were not evaluated in the actual competition and which, to the best of our knowledge, have not yet been directly compared. Second, our target treatment effect parameters are different, as Dorie et al. (2019) focuses on the *average treatment effect on the treated.* Third, while the competition focuses on the overall performance of the methods across all simulation scenarios, we provide more information for the applied researcher by disentangling the effect of the various empirically relevant data features under consideration, on the performance of the methods. Another recent simulation paper on causal machine learning methods is Knaus et al. (2018). The methods evaluated in Knaus et al. (2018) and in this paper are in general different. Our study overlaps with theirs only in terms of the Causal Forest, and the DR MOM method with Lasso and Random Forest. However, in this paper, we extend their DR MOM approach with ML methods not considered in their analysis, i.e., with BART, Boosting, Neural Net and Trees. Furthermore, while Knaus et al. (2018) consider DGPs with and without selection into treatment and different sample sizes, our focus is on data features such as: nonlinearity, overlap, percentage of treated, alignment and heterogeneity. Finally, Knaus et al. (2018) are interested in the finite sample performance of point estimates and report results on the mean squared error, absolute bias and standard deviation. In contrast, we provide more information on the performance of the methods by also computing the confidence interval coverage rates and interval lengths. Other notable simulation analyses containing some of the methods under consideration in this paper but with a smaller set of DGPs than in this study can be found in: Carvalho et al. (2019), Hahn et al. (2019), Jacob (2021), McConnell and Lindner (2019), Wendling et al. (2018).

Other causal machine learning methods with established theoretical properties recently developed in the econometrics literature include: Athey and Imbens (2016), Athey et al. (2018), Colangelo and Lee (2020), Farrell et al. (2021), Semenova et al. (2018). Some of these extend the main methods that we are focusing on. Given the computational costs, we do not include these methods in our analysis, but choose to focus on the ones that received the most attention so far from the empirical economics literature.

The next section presents our simulation framework with the data features relevant for causal inference present in the DGPs, and the calibration of simulations to real data. Section 3 describes the methods used in our evaluation, as well as our newly introduced methods. Section 4 summarizes the performance results on the ATE, while Section 5 presents the performance results on heterogeneous treatment effects: the GATEs and the ITEs. Section 6 performs a sensitivity analysis of the main results to tuning parameter choices. Finally Section 7 concludes.

2 Simulation Framework

We revisit the setup of the 2016 Atlantic Causal Inference Competition by Dorie et al. (2019). The competition was specifically designed for a fair comparison of various causal machine learning methods from the statistics and machine learning literature. The researchers who designed the data generating processes on which the methods were evaluated, were different from the researchers who submitted the competing methods, which ensured a more equitable comparison of methods. Competitors were informed about the following: the data is an observational study with a continuous outcome, a binary treatment indicator, 58 covariates but not all covariates are confounders; the observations are identically and independently distributed; the methods are tested on 77 different simulation settings, with 100 independent replications for each scenario, resulting in 7700 different data realizations; the causal estimand of interest is the effect of the treatment on the treated; the data presents a varying degree of: nonlinearity, percentage of treated, overlap, alignment, treatment effect heterogeneity, and magnitude of the treatment effect. In total, 30 different methods entered the competition.[3]

2.1 Calibration to Real Data

We base our simulation study on the data set used in the 2016 Atlantic Causal Inference Competition. This is a publicly available data set from the Collaborative Perinatal Project (Niswander and Gordon (1972)) on pregnant women and their children between 1959 and 1974. It contains data on 55,000 pregnancies each with over 6,500 variables. The purpose is to study the causal factors of developmental disorders.

Similarly to Dorie et al. (2019), we aim to calibrate our simulations to a real-world data set, with a research question and a list of covariates close to what could be analyzed by an empirical researcher. Choosing the covariates from a real data set has the advantage that it allows the incorporation of plausible types of variables and a natural association

[3] See Dorie et al. (2019) for a description of these methods.

between the covariates. The outcome and the treatment variables are however simulated, to enable modifications of different data features of interest in a measurable way. The relationship between these data features and the performance of the methods can then be analyzed.

We consider covariates that could have been plausibly selected for a twin study on the impact of birth weight on the child's IQ. This leads to 4802 observations and 58 covariates, out 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 5 in the Appendix.

2.2 Data Generating Processes and Data Features

The methods are compared under 77 simulation settings designed to capture various complex scenarios. These are the same 77 data generating processes used in Dorie et al. (2019) and represent types of data features typically encountered in real-data applications. The reason we use the same simulation scenarios as in Dorie et al. (2019) is twofold. First, it is a well-known and comprehensive simulation paper on which a variety of causal machine learning methods from the statistics and machine learning literature have been already evaluated. Second, it captures well-thought "data complications" commonly encountered and important for causal inference. Each setting consists thus of a unique combination of the following criteria (or knobs), relevant for causal inference questions: 1) degree of non-linearity in the outcome and treatment models, 2) degree of percentage of treated, 3) degree of overlap, 4) degree of alignment and 5) degree of treatment effect heterogeneity. We give a full overview of the 77 scenarios and the considered knobs in Table 6 in the Appendix.[4] In what follows we describe the considered DGP knobs.

Degree of nonlinearity. Nonlinearities are included in both the outcome and the treatment models. The outcome equation specifies the relationship between the outcome (or response) variable of interest and the controls, while the treatment equation keeps track of confounding and models the relationship between the treatment variable and controls. The covariates are first passed through a transformation function, and then added or multiplied. For example, with two covariates one can have:

[4] Note that as in Dorie et al. (2019), we do not address issues related to: non-binary treatment, non-continuous outcome, non-iid data, varying data sizes or number of covariates, covariates with measurment erros. These are left for future research.

$f(x_i) = f_1(x_{i1}) + f_2(x_{i2}) + f_3(x_{i1})f_4(x_{i2})$, where x_{ik} represents the k^{th} covariate for the i^{th} individual. The functions $f_j(\cdot)$ can consist of up to third-order polynomial terms, up to three-way interactions between covariates, indicator or step functions, or can be simple linear functions. We consider additive models for the treatment equation, however, when simulating the outcome, we also allow that the sum of $f_j(\cdot)$ be passed through a link function, such as $h(x_i) = exp(f_1(x_{i1}) + f_2(x_{i2}) + f_3(x_{i1})f_4(x_{i2}))$, leading to highly nonlinear outputs. To measure the degree of non-linearity for a specific data set, we can compute Pearson's R^2 from regressing the outcome variables Y on the non-transformed covariates X. In our 7700 different data sets, Pearson's R^2 is between 0.02 and 0.93, with quartiles of 0.26, 0.37, and 0.48.

Percentage of treated. This knob indicates the share of observations receiving the treatment. It ranges from 35% (*low* setting) to 65% (*high* setting).

Overlap. We want to analyze the impact of having control observations that are dissimilar from the treated observations in terms of their confounders. We have again two settings: *full* overlap and *penalized* overlap. *Full* overlap indicates that the propensity score is bounded away from zero and one. In the *penalized* overlap setting, a penalty term is added to the treatment assignment mechanism which forces observations from a particular area of the covariate space to have a propensity score of zero (i.e., these observations are excluded from the treated population regardless of having a high propensity score). The penalization is done using indicator functions that prevent observations with extreme values on several randomly chosen covariates to receive the treatment, see Dorie et al. (2019).

Alignment. Alignment refers to the degree to which the treatment and outcome equations share the same confounding terms. Only those confounding terms that appear in the DGPs of both models are able to cause bias, if they are omitted from the estimation procedure. In general, lack of alignment can create difficulties for methods that favour confounding terms that appear in either the treatment or outcome equations, but not both. The variation of alignment is achieved by specifying a marginal probability that a term in the treatment equation is also in the outcome equation. We have a probability of 0.25 and 0.75 for the *low* alignment and the *high* alignment case, respectively.

Heterogeneity. The knob heterogeneity controls the number of terms interacting with the treatment. *None* implies that the treatment effect is constant conditional on the covariates; *low* means that the treatment is interacted with three of the terms in the response model; and *high* signifies six interactions.

2.3 Performance Measures

We perform 100 simulation repetitions for each of the 77 different scenarios, leading to 7700 data realisations.[5] In our simulation study, the estimands of interest are the average treatment effect (ATE), the group average treatment effects (GATEs) - the most affected and least affected 20%, and the individual treatment effects (ITEs).

As global summaries of performance, we report the bias, the root mean squared error (RMSE), the interval coverage, the average interval length and the precision in estimation of heterogeneous effects (PEHE). Interval coverage indicates the percentage over all data sets that the reported interval covers the true estimand. Considering the trade-off that can exist between interval coverage and interval length, we also report the average interval length. For the individual treatment effects we compute the PEHE (Hill, 2011). This is basically the average across all data sets of the root-mean-squared error between individual level treatment effect estimates and their true values.

3 Causal Machine Learning Methods

In this section, we provide a brief description of the causal machine learning methods used in our simulation study. We then describe our newly constructed causal machine learning methods, based on BART. The aim is to familiarize the reader with these methods without going into technical details.

3.1 Double/Debiased Machine Learning (DML)

The Double/Debiased Machine Learning (DML) introduced in Chernozhukov et al. (2017) and Chernozhukov et al. (2018a) provides consistent estimation and valid inference for causal effects of interest, such as the average treatment effect (ATE), the average treatment effect on the treated (ATTE) and the local average treatment effect (LATE), in the presence of high-dimensional nuisance parameters estimated with machine learning

[5] Note that for the analysis by knobs, the performance will not be assessed across all 77 scenarios, but fewer, depending on how many simulation scenarios contain a certain knob; see Sections 4 and 5.

methods.

In this paper, we consider the estimation of the ATE from the following model

$$Y = g_0(D, X) + U \tag{1}$$

$$D = m_0(X) + V \tag{2}$$

with $E[U|X, D] = 0$ and $E[V|X] = 0$. The function $g_0(X)$ relates the high-dimensional vector of covariates X and the binary treatment $D \in \{0, 1\}$ to the outcome Y, while the function $m_0(X)$ relates X to the treatment D. Note that D is not additively separable. The disturbances of the two equations are denoted by U and V. The first equation is the main equation of interest. In the second equation, the propensity score $m_0(X)$ is zero, in the case of randomized control trials, but it is different from zero in observational studies. Both nuisance functions $g_0(X)$ and $m_0(X)$ are unknown, high-dimensional and are estimated with ML methods.

The ATE parameter of this model θ_0 is expressed as

$$\theta_0 = E[g_0(1, X) - g_0(0, X)].$$

In order to estimate θ_0, the DML method employs moment condition of the type

$$E\psi(W; \theta_0, \eta_0) = 0 \tag{3}$$

where $W = (Y, X, D)$, and η is a nuisance function consisting of g and m, with η_0 the true value of η. The score function $\psi(W; \theta, \eta)$ has to be chosen such that it satisfies the so called Neyman orthogonality introduced in the contributions of Neyman (1959) and Neyman (1979). The estimation of θ_0 is based on the empirical analogue of (3) where η_0 is replaced by $\hat{\eta}_0$. The Neyman orthogonality property ensures that the moment conditions used to identify θ_0 are locally insensitive to this replacement so that one can use noisy estimates of the nuisance parameter η, without strongly violating the moment conditions.[6]

When estimating the ATE, one can use the scores of Robins and Rotnitzky (1995), which possess the property of Neyman orthogonality and satisfy the identification condi-

[6] See e.g., Chernozhukov et al. (2018a) for the formal condition of Neyman orthogonality.

tion in (3),

$$\psi(W;\theta,\eta) = (g(1,X) - g(0,X)) + \frac{D(Y - g(1,X))}{m(X)} - \frac{(1-D)(Y - g(0,X))}{1 - m(X)} - \theta,$$

where $\eta(X) = (g(0,X), g(1,X), m(X))$ and the true value of η is $\eta_0 = (g_0(0,X), g_0(1,X), m_0(X))$. Employing Neyman orthogonality conditions is key in overcoming *regularization bias* and developing valid inference procedures for θ_0. Bias due to regularization naturally arises when trying to estimate the nuisance parameter η_0 with some ML method followed by estimation of θ_0 with OLS. The bias induced by regularization and shrinkage of the less important coefficients to zero when estimating η_0, transfers to the parameter of interest θ_0. The issue is very similar to the omitted variable bias problem. Neyman orthogonal scores are not sensitive to biased estimation of η_0, and thus do not violate the moment conditions.

The second bias that the DML method overcomes is *bias due to overfitting.* Bias due to overfitting can arise for instance when $\hat{g}_0$ is overfit, and thus it will mistakenly pick up some of the noise U. Consequently, if U and V are correlated, the estimation error in $\hat{g}_0$ will be correlated with V. In order to break this correlation, and overcome bias due to overfitting, Chernozhukov et al. (2018a) propose sample splitting. This means that the data is split in a main and an auxiliary subsample. The nuisance functions m_0 and g_0 are estimated on the auxiliary sample, while θ_0 is obtained based on the main sample. For more efficiency, one can switch the role of the main and auxiliary samples and average the results. Moreover, one can also partition the full sample n into k-folds, where n/k is the size of a fold. Each fold is then successively taken as the main sample, while its complement is the auxiliary sample. Finally, the estimates are averaged over the k folds. The splitting in folds procedure can be performed say, s times, so that the results are robust to the k-fold partitioning. The final DML estimator is the mean or median over the s splits. In this paper, we work with the median estimates. Given that we consider 77 different simulation scenarios, each over 100 replications, we choose to use 2 folds and 10 splits for the DML method.

3.2 Doubly Robust Modified Outcome Method (DR MOM)

Knaus et al. (2018) describes a generic machine learning approach to estimate conditional average treatment effects (CATEs) up until the individual level based on the doubly robust estimator of Robins and Rotnitzky (1995),

$$Y_{DR} = (g(1, X) - g(0, X)) + \frac{D(Y - g(1, X))}{m(X)} - \frac{(1 - D)(Y - g(0, X))}{1 - m(X)}. \tag{4}$$

Since the conditional average treatment effect $\tau(x) = E[Y_{DR}|X = x]$, one can estimate the CATEs from the regression of the modified outcome Y_{DR} on the covariates X. In practice, the researcher starts by estimating $g(1, X)$, $g(0, X)$ and $m(X)$ by a machine learning method of choice, estimates of which are then plugged in into (4). Similarly to the DML, we use the plug-ins of the cross-fitted estimated nuisance parameters, however the method is not repeated over s splits. While the asymptotic properties of E$[Y_{DR}]$ as estimator of ATE are well understood, there is no asymptotic theory currently available for for more granular heterogenous effects using this approach.

3.3 Causal Forest (CF)

Wager and Athey (2018) extend the random forest algorithm used for prediction to the problem of treatment effects estimation for different subgroups. Their method, called the causal forest, uncovers heterogeneity in treatment effects by searching over a high-dimensional function of covariates rather than a few interaction terms. Wager and Athey (2018) establish asymptotic normality and consistency results for random forests which are then extended to the causal setting. For valid inference, a consistent estimator of the asymptotic variance is also proposed.

We now describe the general idea of the causal forest. The algorithm starts with drawing a sub-sample from the full sample of observations, without replacement. This whole sub-sample constitutes the root node. The default fraction of the sub-sample is half from the whole sample, but one can change this value: a smaller size for the sub-sample will decrease dependence across trees, but will increase variance. The sub-sample is then further split in half to form a training sample and an estimation sample. The role of the training sample is to define the structure of the tree, while the estimation sample is used to estimate treatment effects in each node of the tree. Using these different samples for different purposes helps with reducing bias from overfitting. Athey and Imbens (2016) and Wager and Athey (2018) call this property "honesty".

Next, a number of covariates are randomly selected to split on, at each split. Using the training sample, for each value of each selected covariate, candidates splits are formed based on the current value of the covariate. In the case of the causal random forest, the goodness of split is given by the amount it increases heterogeneity in a quantity of interest. Instead of minimizing the usual MSE, as in the case of the random forest, Wager and Athey (2018) propose minimizing an objective function that penalizes splits which increase within-node variance but rewards splits which increase the variance of treatment effects across nodes. This criterion function is one of the main differences between the causal forest used for treatment effect estimation and the random forest used for prediction. When computing treatment effects, minimizing a criterion function based on the MSE (obtained as the sum of the squared differences between the outcomes of each observation from a node and the mean of these observation in that node) is not possible, as any individual observation is either treated or not. For each of the new child nodes, the algorithm repeats the splitting procedure until the number of observations in a node continues to be larger than a minimum number of control and treated units. When no more splits can be performed the structure of the tree is defined.

In the next step, the observations from the estimation sample are sorted in the same tree structure as the one obtained from the training sample, based on the values of their covariates. Then, the treatment effect in each node is computed as the mean outcome difference between treated and control units in a node. Finally, going back to the full sample, one examines in which node each unit belongs (again based on the values of their covariates) and the predicted treatment effect assigned to each unit will be the treatment effect of the node where the unit belongs. Estimates obtained in this way, from a single tree, can have a high variance. Thus, the whole procedure is repeated a number of times, say B, leading to B sub-sample and B trees which will form a causal forest. The final predicted treatment effect of each observation will be obtained as the average of predicted treatment effects of that observation across the B trees. To compute the ATE we use the overlap-weighted estimator proposed by Li et al. (2018) which performs better in the case of limited overlap. The GATEs for the least affected and most affected groups can be computed as averages[7] of the top and bottom 20% of individual treatment effects, averaged over the trees. One should be cautious though with the interpretations of the

[7] These averages are computed again with the estimator proposed in Li et al. (2018).

results on GATEs as computed here, because valid statistical inference for GATEs via the Causal Forest is not yet available.

The researcher needs to specify values for the number of trees, the number of covariates considered when splitting, and the minimum number of control and treated units in each nodes. A higher number of trees makes the treatment effect predictions vary less across the trees but increases computation time. The number of trees should be grown in proportion to the number of observations. The number of covariates considered for a split is typically set to $\lfloor P/3 \rfloor$, for regression problems and to $\sqrt{P}$ for classification problems, where P is the total number of covariates. A smaller number of minimum treatment and control units in a node increases the variance as the treatment effect in a node is estimated with fewer observations. A higher number will produce less heterogeneity as the nodes are larger and the tree is less deep. The optimal values for these parameters, as well as for some other parameters, (see the documentation of the **grf** package in **R** for details on other parameters) can be tuned by cross-validation. In this paper, the number of trees is set to 2000, while all other parameters are tuned within the **grf** package.

3.4 Generic Machine Learning (GML)

The generic machine learning method (GML) is useful for uncovering heterogeneity in the treatment effect, computing the treatment effects in different subgroups such as the most or the least affected groups, and giving indications about the covariates that are correlated the most with this heterogeneity.

Consider an outcome variable Y, a binary treatment D and a vector of covariates X. Let $b_0(X) = E[Y(0)|X]$, and $s_0(X) = E[Y(1)|X] - E[Y(0)|X]$ where $b_0(X)$ is the baseline conditional average function and $s_0(X)$ is the conditional average treatment effect (CATE). The GML method randomly splits the data into an auxiliary subsample and a main subsample. Using the auxiliary sample, a ML estimator (called proxy predictor in Chernozhukov et al., 2018b) is computed for the baseline conditional average and the conditional average treatment effect. The ML estimator can use any of the standard ML methods: Lasso, Elastic Net, Random Forest, Neural Network, thus the name *generic*. These ML estimators are possibly biased, however, they are only used as approximations to make inference on *features* of the CATE and not the CATE itself. These features of interest are: 1) the best linear predictor of the heterogeneous effects (BLP), 2) the group average treatment effects (GATEs), and 3) the average characteristics of the units in the most and least affected groups, or classification analysis (CLAN).

In the next step, for each observation from the main subsample we compute the predicted baseline effects, $B(X)$ and the predicted treatment effects, $S(X)$, the latter being computed as the difference between the predictions of the treatment group model and predictions of the control group model. Then, using the main sample, the so called BLP parameters (β_1 and β_2) are obtained from the following weighted OLS regression, with weights $1/(p(Z)(1-p(Z))$:

$$Y = \alpha' Z + \beta_1(D - p(X)) + \beta_2(D - p(X))(S(X) - \overline{S(X)}) + \epsilon. \tag{5}$$

In equation (5), $p(X) = P[D = 1|X]$ is the propensity score[8], $\overline{S(X)}$ is the average of the predicted treatment effect estimates on the main sample, and $Z = [1, B(X)]$, the control $B(X)$ being included to improve efficiency. Note that the interaction term $(D - p(X))(S(X) - \overline{S(X)})$ is orthogonal to $(D - p(X))$ and to all other regressors that are functions of X. The parameter β_1 quantifies the average treatment effect, while β_2 measures how well the proxy predictor approximates heterogeneity. One can test for heterogeneous treatment effects by testing $H_0 : \beta_2 = 0$.

Subsequently, to compute the group average treatment effects (GATEs), we sort the observations from the main sample in groups: $G_1, G_2, \ldots, G_L$. For example, G_1 can contain the units with the lowest 20% treatment effects, while G_5 can contain the units with the highest 20% treatment effects. Then, the group average treatment effects are given by the coefficients γ_l in the weighted regression run on the main sample

$$Y = \alpha' X_1 + \sum_{l=1}^{L} \gamma_l (D - p(Z)) \cdot 1(G_l) + \nu. \tag{6}$$

The weights in (6) are the same as in (5). The indicator function $1(G_l)$ takes values of one when a unit is in the group l. Note that when $\gamma_L - \gamma_1$ is significantly different from zero, it indicates that there is treatment effect heterogeneity between the most and least affected groups. Finally, one can also compute average characteristics of the most affected and least affected groups, i.e., $\delta_1 = E[n(Y, X)|G_1]$ and $\delta_L = E[n(Y, X)|G_L]$, where $n(Y, X)$ is

[8] Note that, since our simulation design is based on an observational study, we also estimate the propensity score with ML methods.

a vector of characteristics of an observation.

In order to account for the uncertainty introduced by the random partitioning of data, the final estimates of the GML method are obtained as the median estimates over the different data splits. Similarly to DML, given that we already have 77 different simulation scenarios, each with 100 replications, we set the number of splits to 10. The confidence intervals are constructed as the medians of the lower and upper bounds over the splits. The price of splitting uncertainty is that the nominal level of confidence intervals as well as of p-values (computed as median over splits) are adjusted from $1-\alpha$ to $1-2\alpha$, where α is the nominal level.

3.5 BART and BART MChains for Treatment Effects

The Bayesian Additive Regression Trees (BART) method developed in Chipman et al. (2010) for prediction purposes and popularized in Hill (2011) for treatment effect estimation, is a sum-of-trees method which estimates a model for the outcome Y as $Y = g(d,x) + \varepsilon$, where d is the treatment, x are the confounding covariates and ε are iid $N(0,\sigma^2)$. When the purpose is treatment effect estimation, the treatment variable d is considered a splitting variables as all the other covariates.

Intuitively, the BART method consists of a sum-of-trees model and a regularization prior on the parameters of the model. Similarly to Boosting, the method computes the fit based on the first tree and then subtract it off from the outcome, forming the residuals. The next tree is then fitted on these residuals. The process is repeated a number of B times. The regularization prior avoids overfitting, by constraining the fit of each tree, such that each tree explains a different minor portion of $g(\cdot)$. Without this regularization, individual tree components would become too influential, limiting the advantages of the additive representation in terms of functional approximation.

Formally, we have

$$g(d,x) + \varepsilon = \sum_{j=1}^{B} h(d,x;T_j,M_j) + \varepsilon$$

where B is the number of trees, T_j is one of the binary trees and $M_j = (\mu_1, \mu_2, \ldots, \mu_b)$ is the set of parameters of the b terminal nodes in each tree – the parameter of a terminal node being the mean response of the subgroup of observations that fall in that node. The function $h(d,x;T_j,M_j)$ gives then the μ associated with a particular terminal node in a particular tree.

BART treats (T_j, M_j) and σ as parameters and sets a prior on them, the posterior being computed with Markov chain Monte Carlo (MCMC). The prior has three components: 1) a prior on the tree structure – preference is given for trees with only a few terminal nodes; 2) a prior for the values in the terminal nodes – which shrinks each M_j towards zero (note that the response variables is centered) limiting the effects of the individual trees by keeping them small; and 3) a prior for σ chosen based on the residual standard deviation from a simple least squares regression.[9] In this paper, we use the default prior settings provided by Chipman et al. (2010), as well as the recommended value of 200 for the number of trees, B.

At each iteration of the MCMC algorithm, the pair (T_j, M_j) and σ are redrawn. Let $T_{(j)}$ be the set of all trees in the sum-of-trees except T_j. Similarly define $M_{(j)}$. The algorithm involves B successive draws of (T_j, M_j) conditional on $(T_{(j)}, M_{(j)}, \sigma)$, followed by a draw of σ from the full conditional $(T_1, \ldots, T_B, M_1, \ldots, M_B)$. Note that the B draws of (T_j, M_j) given $(T_{(j)}, M_{(j)}, \sigma)$ are equivalent to B draws of (T_j, M_j) given (R_j, σ), where R_j are the residuals resulted after fitting tree T_j. At each iteration, each T_j can grow or become smaller. The contribution of one particular tree is not identified, as one can switch the pair (T, M) with another, without changing $g(\cdot)$. This lack of identification gives flexibility to the method to reallocate the local fit from one tree to another. Only the parameter σ is identified. Thus, one can check the convergence of the chain by plotting draws of σ. In this paper, to ensure convergence we adopt the default recommendations in Chipman et al. (2010) of 100 burn-ins and 1000 MCMC iterations.

The MCMC algorithm induces a sequence of sum-of-tree functions

$$g^*(\cdot) = \sum_{j=1}^{B} h(\cdot; T_j^*, M_j^*),$$

for the sequence of draws $(T_1^*, M_1^*), \ldots, (T_B^*, M_B^*)$, which converges to the posterior distribution of the true $g(\cdot)$. Bayesian inferential quantities can then be approximated based on the sequence of draws, say $g_1^*, \ldots, g_K^*$. When interested in treatment effects estimation, one can compute the ITE - at draw k as $\tau_i^k = g_k^*(1, x_i) - g_k^*(0, x_i)$, for all observation $i = 1, \ldots, N$. Then, the average treatment effect at draw k is computed as

[9] See Chipman et al. (2010) for further detail on the priors and the hyper-parameters.

$\beta^k_{ate} = \frac{1}{N}\sum_{i=1}^{N} \tau_i^k$. Iterating over all $k = 1, \ldots, K$ draws, yields the posterior distribution. The ATE can be computed as $\beta_{ate} = \frac{1}{K}\sum_{k=1}^{K} \beta^k_{ate}$, while the GATEs can be obtained as $\beta_g = \frac{1}{K}\sum_{k=1}^{K} \beta^k_g$, where $\beta^k_g = \frac{1}{N_g}\sum_{i \in G}^{N_g} \tau_i^k$, with N_g being the number of observations in a group, and G a group set such as the most affected or the least affected group. Posterior intervals at $(1-\alpha)\%$ can then be obtained based on the upper and lower $\alpha/2$ quantiles, of the set of draws, for the treatment parameter of interest.

The BART method discussed so far usually runs with a single chain. Dorie et al. (2019) propose to combine the results from several chains with distinct starting points, calling it BART MChains. This method ends up to be one of the top performing methods in the Atlantic Causal Inference Conference Competition. In this paper, we choose to implement BART MChains with ten chains as in Dorie et al. (2019).

3.6 Newly Constructed Causal Machine Learning Methods

Given the promising results obtained by BART based causal machine learning methods in the Atlantic Causal Inference Conference Competition (Dorie et al. 2019), we extend the Double Machine Learning method, the Generic Machine Learning method and the Doubly Robust Modified Outcome method with BART. The nuisance function are thus estimated in each case with BART. The Double Machine Learning, the Generic Machine Learning and the Doubly Robust Modified Outcome method combined with BART turn out to be among the top performing methods in our analysis surpassing the other ML combinations in most of the cases. [10]

4 Results on Average Treatment Effect

We start by analyzing the *overall* performance of the methods, in terms of the average treatment effect, with respect to *all* criteria/data features described in Section 2. Then, we analyze the impact of varying each criteria (degree of nonlinearity, percentage of treated, overlap, alignment and heterogeneity), separately.

[10] The **R** code for these new methods is available online on GitHub under https://github.com/cpwirths/CML_ATE_HTE. We also provide here the replication code for this paper.

4.1 ATE: Overall Performance

Figure 1 presents the overall performance of the methods. It displays the bias and RMSE in Panel A, and coverage and interval length in Panel B, across all 7700 data realisations. The horizontal lines reflect the desired theoretical value for the bias, RMSE and the coverage. Several points are worth noting. First, the method that has the best overall performance in terms of all evaluation criteria (bias, RMSE, coverage and interval length) is BART MChains. The bias is -0.001, the RMSE is 0.02, the coverage is 89% and the interval length is 0.04. Second, the newly introduced methods, DML BART and GML BART tend to outperform the other ML techniques within the DML and GML frameworks, respectively. In terms of RMSE and coverage, DML BART and GML BART come second to BART MChains, while in terms of bias and interval length, they are also within the very top performing methods. Third, the performance of DML and GML is similar when compared along the same individual ML method. In terms of the *original* ML methods used within DML and GML (i.e., not combinations with BART), Boosting and Random Forest seem to give the best performance, while Lasso and Neural Net lead to the worst performance. Lastly, the Causal Forest method presents a particularly low coverage (51%), while the OLS method clearly has the overall worst performance.

Notice further that the average bias is negative for most methods, indicating that the treatment effect had a positive effect and most methods shrink their estimates towards zero. While most CML methods perform well in terms of bias and RMSE, only BART MChains, DML BART and GML BART have a coverage above 80%, see Table 7 in the Appendix for exact values. Also, none of the methods reaches nominal coverage. BART MChains uses several Markov Chains with distinct starting points and performs better than the simple BART used within the DML and GML frameworks.

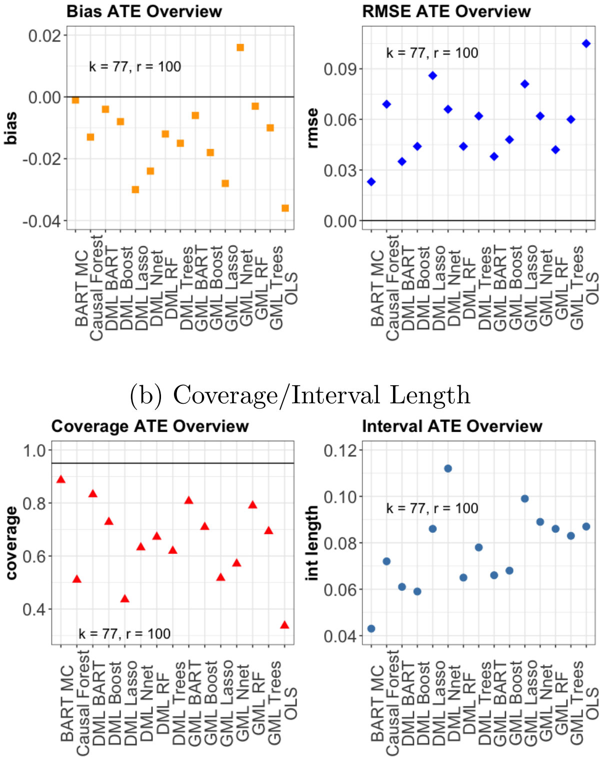
To gain deeper insights into the performance of the individual ML methods within the DML and GML frameworks, we have a closer look at their performance in predicting the nuisance functions. Figure 2, plots the RMSE of the response functions $E[Y|D = 1, X]$ and $E[Y|D = 0, X]$ and the Brier score[11] of the treatment function $E[D|X]$. The ML methods present substantial differences when predicting the response functions compared to when predicting the treatment function. In the case of the response model, DML BART

[11] Here, the Brier score is mean square error between the predicted propensity score and the observed binary treatment variable, where values of zero are the best score achievable.

and GML BART perform the best, followed by DML Boosting and GML Boosting - these all grow a sequence of trees using a weak learner approach. The next best performing method is the DML Forest and GML Forest which grow a multitude of trees to avoid overfitting and to lower the variance compared to the single tree approach. In general, Tree-based structures appear to be more appropriate, given that even a single tree approach performs better than the Lasso or the Neural Net.[12]. In the case of the treatment function, the ML methods perform similarly, although the differences between them are less pronounced. The superior performance of the BART and Boosting methods observed in Figure 1 seems to be explained by the ability of these methods to more flexibly model the response surface compared to the other methods (see Dorie et al. 2019 for a similar observation on the methods that performed the best in the Atlantic Causal Inference Competition).

Figure 1: Overall performance: ATE, averaged across all 77 simulation scenarios

(a) Bias/RMSE

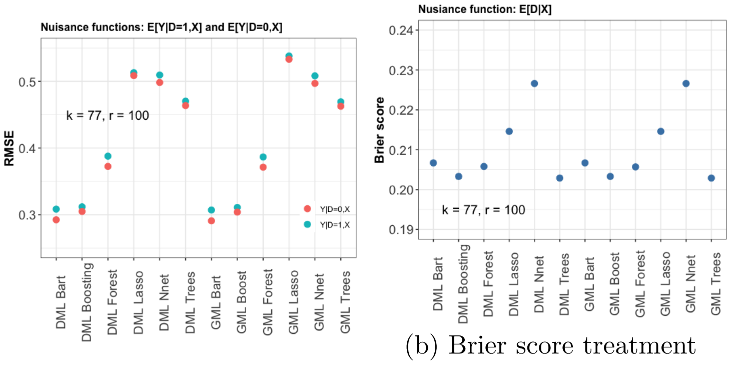


(b) Coverage/Interval Length

Note: The figure display the performance of all methods in estimating the ATE, with results averaged across all $k = 77$ simulation settings, with $r = 100$ replications per setting. Thus, in total we consider 7700 data sets. In Panel A, squares reflect bias, diamonds Root Mean Square Errors, while in Panel B, triangles reflect coverage and circles interval lengths.

[12] Note that we keep the Neural Net with one hidden layer and two neurons, while the Lasso includes polynomial terms up to third order, as it has been originally implemented in Chernozhukov et al. (2018a).

Figure 2: Performance nuisance parameters



(a) RMSE response functions

(b) Brier score treatment function

Note: The figure displays the performance of all methods in predicting the nuisance parameters with results averaged across all $k = 77$ simulation settings, with $r = 100$ replications per setting. Panel A gives 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 treatment function ($E[D|X]$).

4.2 ATE: Performance by Different Criteria

Besides studying the overall performance of the methods, we are now interested to analyze the impact of each criteria/data feature separately. To this end, we select simulation scenario number 27 from Table 6 in the Appendix as the benchmark. This setting has a polynomial treatment model, low percentage of treated, full overlap, step response model, low alignment and low heterogeneity. Then, we alter each knob one at a time, as displayed in Table 1 to study the impact of the various data features on the causal machine learning methods.

Our benchmark, scenario number 27, is one of the simplest available scenarios that permits to alter each criteria/knob one at a time, and still be able to find the altered scenario within the possible settings of Table 6 in the Appendix. Note for example, that by choosing scenario number three as the benchmark (which has linear functions for both the treatment and outcome functions), we are not able to perform a similar change for each criteria as in Table 1. For further insights, we also make comparisons by aggregating all simulation results from high versus low percentage of treated, full versus penalized overlap, high versus low alignment, high versus low heterogeneity, polynomial versus step treatment model and step versus exponential response model.

Table 1: Simulation settings used for ATE performance analysis by different criteria

scenario	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**

Notes: The table displays the simulation scenarios chosen to analyze the performance of the methods under different data features. Simulation scenario number 27 is our benchmark. We then alter each criteria one at a time.

When we change the functional form of the treatment model from *polynomial* to *step*, but keep all the other data features as in our benchmark scenario, the Lasso based methods perform worse. This can be seen in Table 2, as we move from the benchmark setting to setting number 55, or even more clearly from Figure 5 in the Appendix. In Figure 5, we compute the evaluation measures over all $k = 39$ scenarios from Table 6 in the Appendix that have a polynomial treatment function and all $k = 32$ scenarios with a step function for the treatment equation. In contrast, the Tree-based methods show improvements when we move from a *polynomial* to a *step* function. These results are intuitive as Lasso is well suited to capture polynomial terms, but cannot incorporate step functions. On the other hand, Tree-based methods, employ splitting rules based on cutoff points that divide the covariate space, in order to minimize a certain criterion function. Since different values are predicted in the sub-regions of the covariate space, discontinuous functional forms, such as step functions, are easier captured by Tree-based methods. Altering the functional form of the response model from *step* to *exponential* (scenario 31) does not result in notable changes, except in slightly larger interval lengths. Notice further that OLS is consistently outperformed by the causal machine learning methods under polynomial/step functions of the treatment model, or step/exponential functions of the response model, in terms of bias, RMSE and coverage. This highlights the usefulness and higher flexibility of modern causal machine learning methods in empirical applications where non-linearities are expected.

For further analysis we also add the case of Figure 7 in the Appendix, where we aggregate the results on all linear versus nonlinear DGPs. When both response and treatment models are nonlinear the OLS presents larger bias, RMSE and interval lengths compared to the case when both response and treatment models are linear. In addition, we notice that the performance of Tree-based methods is negatively affected in the purely linear case as they are not designed to capture linear effects, see also Friedberg et al. (2020).

When we change the *percentage of treated* from low to high (scenario 40), it does not lead to changes in the performance of most methods relative to the benchmark (notably, only the RMSE of the Causal Forest increases from 0.02 to 0.03 and its coverage decreases from 0.87 to 0.80). Figure 8 in the Appendix, where we compare the performance of the methods over all scenarios with low percentage of treated versus all scenarios with high percentage of treated, confirms this remark.

As we move from *full* to *penalized* overlap and violate the common support assumption (scenario 21), the RMSE increases for all methods, while the coverage rates drop for almost all methods. Figure 9 in the Appendix confirms this result. Figure 9 also shows that with full overlap, besides BART MChains, the newly proposed DML BART reaches nominal coverage. Under penalized overlap, BART MChains is overall the best performing method, closely followed by DML BART and GML BART. Notice that although DML Neural Net and GML Random Forest also seem to have a good performance in terms of coverage, this comes at a cost of an increased confidence interval length.

Crump et al. (2006) point out that violations of the overlap assumption can lead to substantial bias and larger variance of conventional estimators of average treatment effects. Moreover, limited overlap can also cause problems for inference due to its detrimental effect on the coverage probability of standard confidence intervals, see for example Rothe (2017). We observe similar effects in the case of causal machine learning methods. Note that when implementing the methods, we already employ versions that try to address the lack of common support problem. In the case of the DML and the GML methods, we trim the observations with predicted propensity scores below 0.01 and above 0.99, at each split. In the case of the CF, we use the weighted estimator introduced by Li et al. (2018), recommended in the case of poor overlap. Finally, in the case of BART MChains, DML BART and GML BART, we can identify areas with penalized overlap, since the standard deviation of the posterior distribution increases in the regions with lack of common support (Dorie et al. 2019). Based on this information we omit observations using a discarding rule suggested in Hill and Su (2013)[13] which has been shown to be robust across various simulation settings.

[13] We used the so-called *1-sd-rule* from Hill and Su (2013).

Table 2: ATE performance analysis by different criteria/data features

Scenario	BART MC	Causal Forest	DML BART	DML Boost	DML Lasso	DML Nnet	DML RF	DML Trees	GML BART	GML Boost	GML Lasso	GML Nnet	GML RF	GML Trees	OLS
Panel A: Bias															
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
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	0.02	0.00	-0.01
40	-0.00	-0.01	-0.00	-0.00	-0.01	-0.01	-0.00	-0.01	0.00	-0.00	-0.01	0.02	0.01	-0.00	-0.01
21	0.00	-0.01	-0.00	-0.01	-0.01	-0.01	-0.00	-0.01	-0.00	-0.02	-0.01	0.03	0.00	-0.00	-0.01
31	0.00	-0.01	-0.00	-0.00	-0.01	-0.01	-0.00	-0.01	0.01	-0.00	-0.01	0.02	0.01	-0.00	-0.01
29	-0.00	-0.01	-0.01	-0.01	-0.04	-0.03	-0.02	-0.03	0.00	-0.01	-0.03	0.01	0.00	-0.02	-0.05
28	-0.00	-0.00	-0.01	-0.01	-0.04	-0.03	-0.02	-0.02	-0.00	-0.01	-0.03	0.01	0.01	-0.01	-0.05
Panel B: RMSE															
27	0.01	0.02	0.01	0.01	0.04	0.03	0.02	0.02	0.01	0.01	0.04	0.04	0.02	0.02	0.07
55	0.01	0.03	0.01	0.01	0.06	0.04	0.01	0.02	0.01	0.01	0.06	0.05	0.02	0.02	0.08
40	0.01	0.03	0.01	0.01	0.04	0.03	0.02	0.02	0.01	0.01	0.04	0.04	0.02	0.02	0.06
21	0.01	0.03	0.03	0.03	0.04	0.04	0.03	0.03	0.03	0.04	0.04	0.04	0.03	0.03	0.05
31	0.01	0.03	0.01	0.01	0.04	0.03	0.02	0.02	0.01	0.01	0.04	0.04	0.02	0.02	0.06
29	0.01	0.05	0.01	0.02	0.06	0.06	0.03	0.04	0.01	0.01	0.05	0.04	0.02	0.04	0.11
28	0.01	0.04	0.02	0.02	0.07	0.06	0.03	0.04	0.01	0.01	0.07	0.05	0.02	0.03	0.12
Panel C: Coverage															
27	0.97	0.87	0.96	0.93	0.74	0.85	0.89	0.86	0.96	0.96	0.84	0.70	0.83	0.88	0.67
55	0.97	0.79	0.98	0.95	0.72	0.86	0.97	0.93	0.98	0.97	0.78	0.60	0.73	0.97	0.69
40	0.98	0.80	0.96	0.92	0.73	0.84	0.87	0.87	0.95	0.95	0.84	0.72	0.83	0.92	0.63
21	0.92	0.68	0.78	0.73	0.70	0.86	0.71	0.77	0.85	0.72	0.80	0.68	0.84	0.84	0.64
31	0.92	0.83	0.99	0.96	0.74	0.92	0.95	0.88	0.92	0.96	0.78	0.61	0.82	0.96	0.65
29	0.98	0.49	0.90	0.68	0.42	0.53	0.63	0.46	0.94	0.87	0.52	0.64	0.87	0.68	0.31
28	0.92	0.59	0.89	0.83	0.38	0.58	0.65	0.69	0.91	0.92	0.57	0.52	0.84	0.81	0.16
Panel D: Interval Length															
27	0.03	0.06	0.05	0.05	0.08	0.09	0.06	0.06	0.05	0.05	0.10	0.08	0.06	0.07	0.08
55	0.03	0.06	0.05	0.05	0.08	0.09	0.05	0.06	0.04	0.04	0.10	0.08	0.05	0.06	0.08
40	0.03	0.06	0.05	0.05	0.08	0.09	0.06	0.06	0.04	0.04	0.10	0.08	0.05	0.07	0.08
21	0.04	0.06	0.05	0.05	0.08	0.12	0.06	0.07	0.06	0.07	0.11	0.10	0.07	0.08	0.09
31	0.03	0.07	0.06	0.06	0.07	0.09	0.07	0.07	0.05	0.05	0.08	0.08	0.06	0.08	0.08
29	0.03	0.06	0.05	0.05	0.07	0.09	0.06	0.06	0.05	0.04	0.08	0.07	0.06	0.07	0.08
28	0.03	0.07	0.06	0.06	0.08	0.10	0.07	0.07	0.05	0.05	0.09	0.08	0.06	0.07	0.09

Notes: The table summarizes the bias, RMSE, coverage rates and interval lengths when estimating the ATE, with 100 replications per scenario.

Increasing the level of *alignment* from low to high (scenario 29) leads to a mixed performance with respect to the coverage rates of average treatment effects. While BART MChains, DML BART, GML BART and GML Random Forest are relatively robust, the coverage rates for the Causal Forest, DML Lasso, DML NNet, DML Trees and GML Lasso decrease substantially (see Table 2). A similar pattern is observed in terms of increased bias, as shown in Figure 10 in the Appendix. Note that in the simulation design, high alignment is achieved by specifying a higher marginal probability that a term in the treatment assignment mechanism is copied to the response function. This inevitably leads to a higher dimension of the confounder space and possibly increased nonlinearity/complexity in the DGP which decreases the coverage rates and increased the bias for some of the CML methods. Furthermore, a high alignment deteriorates the performance of the OLS in terms of bias, RMSE and coverage rates, since the response, the treatment and control variables are modeled in one regression equation, without data-driven variable selection and estimation of the propensity score.

When we increase the *level of heterogeneity* from low to high (scenario 28) most methods perform worse. This is expected as we increase the complexity/nonlinearity of the model. Some methods are more impacted than others. BART MChains, DML BART, DML Boosting, GML BART, GML Boosting, GML Random Forest are more stable in terms of RMSE and coverage rates, while DML Lasso, DML Neural Net, GML Lasso and GML Neural Net are the most affected causal machine leaning methods. In general, when heterogeneity increases, Tree-based methods have a superior performance as they pick up complex interactions by constructions. As expected, OLS is severely affected, as it does not include any interactions with the treatment. Figure 11 in the Appendix which presents results by grouping all scenarios with low and high heterogeneity confirms these remarks.

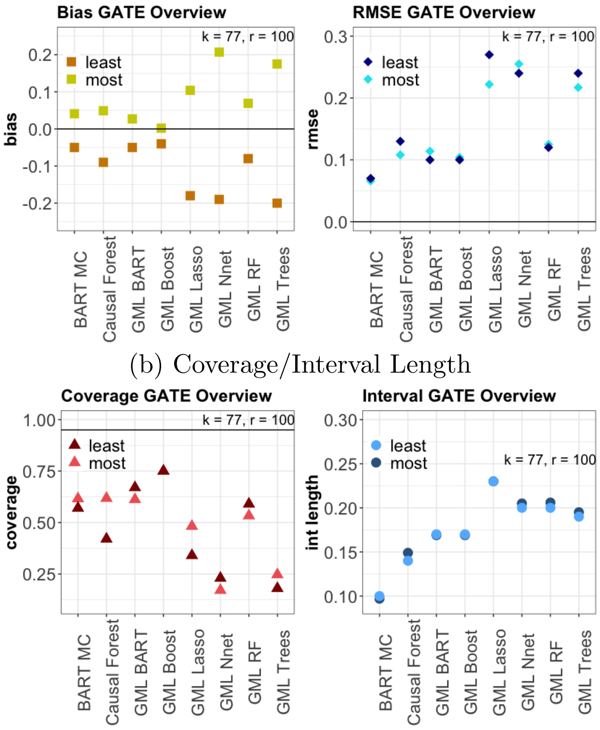
5 Results on Heterogeneous Treatment Effects

In this section, we analyze the performance of the causal machine learning methods when estimating heterogeneous treatment effects. First, we focus on group average treatment effects (GATEs), in particular on the 20% most and 20% least affected groups, oftentimes of interest in policy evaluation. Here we focus on the methods that allow the computation of GATES, i.e., BART MChains, Causal Forest and the GML methods. Then, we assess the performance of the methods when computing individual treatment effects (ITEs). In this case, we work with the methods that allow the computation of ITEs, i.e., BART MChains, Causal Forest and the DR MOM methods.

5.1 GATEs: Overall Performance

We provide an overview of the bias, RMSE, coverage rates and interval lengths across all 77 simulation scenarios, of the most and the least affected group effects in Figure 3 below and in Table 8 in the Appendix.

Figure 3: Overall performance: GATEs, averaged across all 77 simulation scenarios



(a) Bias/RMSE

(b) Coverage/Interval Length

Note: The figure displays the performance of all methods in estimating the GATEs, with results averaged across all $k = 77$ simulation settings, with $r = 100$ replications per setting. Thus, in total we consider 7700 data sets. In Panel A, squares reflect bias, diamonds Root Mean Square Errors, while in Panel B, triangles reflect coverage and circles interval lengths.

The results on GATEs present higher biases, higher RMSEs, lower coverages and wider confident intervals when compared to the overall ATE results. In terms of overall GATEs performance, we distinguish two groups. Ensemble methods that grow a sequence of trees, i.e., BART MChains, Causal Forest, GML BART, GML Boosting and GML Random Forest are the best performing methods. Within this group, the ensemble methods which grow a sequence of trees based on a weak learner approach, i.e., BART and Boosting, perform best. BART MChains has the lowest RMSE and the shortest interval length, while GML Boosting achieves the lowest bias and the highest coverage rate. Furthermore, notice that the Causal Forest ranks among the top performing methods, which was not the case in the ATE analysis. The second group made up of GML Lasso, GML Neural Net and GML Trees performs notably worse. Finally, the advantage of growing multiple trees becomes more evident when estimating heterogeneous treatment effects, since when comparing the scales of the RMSEs in Figure 1 (ATE analysis) and 3 (GATE analysis), it becomes clear that the Generic Tree-based ensemble methods considerably outperform the Generic (single) Tree.

5.2 GATEs: Performance by Different Criteria

We turn now to the analysis of each criteria, separately. For the GATEs analysis we select simulation scenario number 28 from Table 6 in the Appendix as the benchmark. This is similar to the previous benchmark (scenario number 27), with the exception that the level of heterogeneity is set to high, since our main focus here is to evaluate how well the methods are capturing heterogeneity. As previously, we alter each knob one at a time, as displayed in Table 3, to evaluate the performance of the different causal machine learning methods under various data features. For further analysis, we also aggregate all simulation results from high versus low percentage of treated, full versus penalized overlap, high versus low alignment, high versus low heterogeneity, polynomial versus step treatment model and step versus exponential response model.

scenario	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 3: Simulation settings used for GATEs performance analysis by different criteria

Notes: The table displays the simulation scenarios chosen to analyze the performance of the methods under different data features. Simulation scenario number 28 is our benchmark. We then alter each criteria one at a time.

As we alter the treatment model from *polynomial* to a *step* function (scenario 56), Table 4 and Figure 12 in the Appendix reveal that GML Lasso performs worse, while the Tree-based methods perform better, especially in terms of coverage rates. This is consistent with the results on the ATE. When the response model changes from a *step* function to an *exponential* function (scenario 32), the methods perform overall slightly worse.

As we increase the *percentage of treated* from low to high (scenario 41), Figure 14 in the Appendix shows no notable changes. There is only a small performance gap increase (in the RMSE and coverage) between the treatment effect of the most affected group and the least affected group, in the case of Causal Forest.

When moving from *full* to *penalised* overlap (scenario 22), the RMSE of all methods increases. We observe however mixed results when looking at the coverage rates. There are some methods for which the coverage rates actually increase (Causal Forest, GML BART, GML Neural Net, GML RF), but when looking at the interval lengths we observe that these also increase. Overall, taking into account all evaluation criteria, it seems that BART MChains and Causal Forest are the most stable to violations of the overlap assumption.

When changing the level of *alignment* from low to high (scenario 30), the changes in the performance measures are smaller compared to the ATE case, although the general pattern remains the same.

We also make the change from high to low *heterogeneity* (scenario 27). As expected, the RMSEs and biases become overall smaller, while the coverage rates increase and interval lengths decrease. When heterogeneity is high in the data, Tree-based ensemble methods (BART MChains, Causal Forest, GML BART, BML Boosting) perform overall the best.

5.3 ITEs: Overall Performance and Performance by Different Criteria

Individual treatment effects (ITEs) are especially useful, for example, in the fields of personalized marketing or personalized medicine. In this section, we focus on the causal machine learning methods which provide ITEs estimates. Figure 4 presents the performance of the methods in terms of the PEHE when computing the ITEs. The results are averaged across all 77 simulation scenarios with 100 replications per scenario. The best performing method for ITEs estimation is BART MChains, followed by DR MOM Boosting, DR MOM BART and Causal Forest. DR MOM Lasso and DR MOM Neural

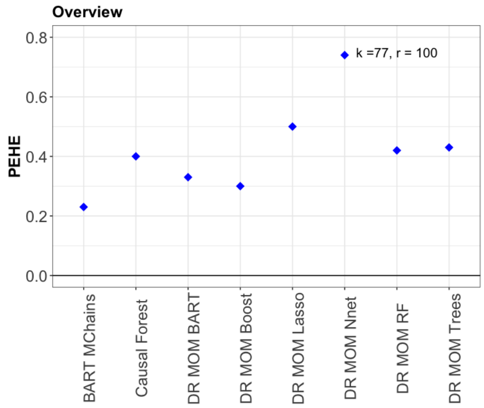
Table 4: GATE performance analysis by different criteria/data features

Scenario	BART MC		Causal Forest		GML BART		GML Boost		GML Lasso		GML Nnet		GML RF		GML Trees	
	most affect.	least affect.	most affect.	least affect.	most affect.	least affect.	most affect.	least affect.	most affect.	least affect.	most affect.	least affect.	most affect.	least affect.	most affect.	least affect.
Panel A: Bias																
28	0.04	-0.04	0.05	-0.08	0.05	-0.05	0.03	-0.04	0.11	-0.19	0.21	-0.19	0.09	-0.09	0.16	-0.18
56	0.04	-0.04	0.06	-0.06	0.05	-0.04	0.02	-0.03	0.09	-0.22	0.20	-0.21	0.09	-0.07	0.14	-0.14
41	0.04	-0.04	0.05	-0.07	0.05	-0.05	0.04	-0.04	0.13	-0.17	0.22	-0.20	0.09	-0.08	0.17	-0.18
22	0.05	-0.05	0.05	-0.08	0.03	-0.05	0.01	-0.04	0.09	-0.30	0.24	-0.20	0.09	-0.08	0.21	-0.20
32	0.05	-0.05	0.09	-0.12	0.06	-0.05	0.04	-0.05	0.13	-0.20	0.24	-0.22	0.12	-0.11	0.20	-0.24
30	0.04	-0.05	0.04	-0.10	0.05	-0.06	0.04	-0.06	0.08	-0.19	0.19	-0.19	0.10	-0.10	0.16	-0.22
27	0.03	-0.03	0.03	-0.05	0.04	-0.03	0.02	-0.04	0.12	-0.14	0.20	-0.15	0.08	-0.06	0.13	-0.16
Panel B: RMSE																
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
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
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
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
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
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
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
Panel C: Coverage																
28	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	0.70	0.65	0.50	0.52	0.62	0.64	0.83	0.86	0.35	0.31	0.21	0.23	0.27	0.51	0.33	0.22
41	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	0.66	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	0.45	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	0.67	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	0.73	0.74	0.71	0.58	0.60	0.68	0.87	0.76	0.52	0.43	0.13	0.27	0.37	0.57	0.27	0.18
Panel D: Interval Length																
28	0.09	0.09	0.13	0.13	0.14	0.14	0.13	0.12	0.20	0.20	0.19	0.19	0.15	0.15	0.17	0.17
56	0.10	0.10	0.12	0.12	0.12	0.12	0.11	0.11	0.24	0.24	0.20	0.20	0.13	0.13	0.15	0.15
41	0.10	0.10	0.14	0.12	0.14	0.14	0.12	0.13	0.20	0.20	0.19	0.19	0.15	0.15	0.18	0.18
22	0.13	0.12	0.17	0.16	0.21	0.20	0.20	0.20	0.31	0.30	0.24	0.24	0.28	0.26	0.21	0.21
32	0.09	0.08	0.16	0.14	0.15	0.15	0.14	0.14	0.21	0.21	0.20	0.20	0.16	0.16	0.20	0.20
30	0.09	0.08	0.14	0.12	0.14	0.14	0.13	0.13	0.19	0.19	0.18	0.18	0.15	0.15	0.18	0.18
27	0.09	0.08	0.12	0.12	0.12	0.12	0.11	0.12	0.22	0.22	0.19	0.19	0.13	0.13	0.16	0.16

Notes: The table summarizes the bias, RMSE, coverage rates and interval lengths when estimating the GATE, with 100 replications per scenario.

Net are at the other end of the performance rank. Notice that the order of performance of the ML methods within the DR MOM approach is similar to the one observed in Figure 2 when estimating the response function.

Figure 4: Overall performance: PEHE for ITEs



Note: The figure displays the Overview of the Precision in Estimating Heterogeneous Effects (PEHE) for the ITEs. The results are averaged across all $k = 77$ simulation settings, with $r = 100$ replications per setting. Thus, in total we consider 7700 data sets.

As in the case of ATE and GATEs, we perform next an analysis by the different data features. For the ITEs, we report the aggregated overview from Figure 18 in the Appendix. We notice that when the overlap assumption is violated or when we increase the level of heterogeneity in the data, the PEHE increases for all methods. In terms of percentage of treated or alignment, the changes in PEHE are very small. Moving from a step function to a polynomial treatment model, or from a step function to an exponential response model, increases the PEHE for most methods. BART MChains remains the best method regardless of which criteria we change.

6 Sensitivity Analysis

The results of Sections 4 and 5 are obtained with the tuning parameters of the individual ML methods taking the values specified in Table 9 in the Appendix. Some of these parameters take default values while other parameters are already tuned with built-in tuning functions available in the **R**-package of the ML method. The results presented in the previous sections might be subject to some of the default values used. In this section we perform a sensitivity analysis to default values. To this end, the parameters which are additionally available for tuning within the *caret* R-package are varied over different parameter combinations. The *caret* package is a general package that among other functionalities can streamline the training process of the different ML methods, permitting the tuning of additional parameters that are not tuned in the ML package.

Since tuning the ML input parameters over a set of varying parameter combinations is computationally intensive, we implement the sensitivity analysis for the main simulation scenarios from Table 1 and Table 3 and not for all 77 simulation settings. Thus, we replicate the results of Table 2, and Table 4 when the values of the tuning parameters (the ones which are available for tuning in *caret*) for Boosting, Random Forest and Neural Net, used within the DML and GML frameworks, are chosen over a set of different values. We omit the DML and GML methods with Lasso and Trees as well as the Causal Forest, since the parameters for these methods were already tuned in Section 4 and 5.[14] We also omit the BART-based methods given that most papers do not tune BART, as cross-validating the prior is hard to justify. In addition, as noted by Chipman et al. (2010) the default BART performs only slightly worse than the tuned BART, but the execution time is much faster. See Table 10 in the Appendix for an overview of the tuning parameters of all methods from the main and the sensitivity analysis.

For each simulation scenario we need to tune over all 100 simulation replications. We do not tune however over each sample splitting step within the DML and the GML methods[15] due to computational costs. For each method, we consider a number of tuning parameter combinations as shown in Tables 11-13 in the Appendix. The optimal values of the tuning parameter are chosen by minimising the aggregated RMSE over the hold-out sample sets via repeated cross-validations. These optimal values are then used as inputs for the ML methods at each simulation replication. The results revisited with the optimal tuning values are given in Table 14 and Table 4.

Comparing the results of Table 14 and Table 2, on the ATE, we notice that the coverage rates improve slightly when the input parameters are tuned. However, our main points on the ATE results continue to hold. On the other hand, the RMSEs, biases and interval lengths are not sensitive to the values of the tuning parameters. When looking at the GATEs analysis and comparing Tables 15 and 4, we notice again that the results are not sensitive to the choice of the tuning parameters and our main points on the GATEs results continue to hold. Given that our results on the ATE and GATEs are not sensitive

[14] Note that for the Trees method some of the tuning parameter stay at their default values as the *caret* package currently does not support their tuning. The same is happening for the minimum node size parameter in the Causal Forest.

[15] Similarly to Sections 4 and 5, we perform 10 sample splits for both DML and GML.

to the tuning parameter choices, we do not consider further sensitivity checks on the ITE necessary.

7 Conclusion

This paper investigates the performance of causal machine learning methods, newly introduced in the economics literature, by revisiting the 2016 Atlantic Causal Inference Competition. The analysis calibrates the simulations to a real-life data set and provides a comprehensive framework to evaluate the estimation of treatment effects with different granularities, across 77 different data generating processes. The focus is on data complications of interest in causal inference such as varying degrees of: nonlinearity in the outcome and treatment equations, overlap, percentage of treated, alignment and heterogeneity of the treatment effect.

When estimating the ATE, the top performing method, overall, is the BART MChains followed by DML BART. After DML BART, the best performing ML methods used within the DML framework, in decreasing order, are Boosting, Random Forest, Trees, Neural Net and Lasso. Although the DML has marginally better results, the performance of the GML is similar. The Causal Forest is not among the best performing methods, especially with respect to coverage rates. All CML methods outperform the OLS estimator, demonstrating their strength in settings characterized by nonlinear response and treatment functions, high-dimensional confounding effects and high heterogeneity.

In terms of heterogeneous treatment effects, when estimating the GATEs, we find that all methods perform worse than in the ATE estimation, as the RMSEs of the best performing methods are higher and the coverage rates are more below the nominal coverage. In the overall analysis, among all methods, the Generic Boosting has the lowest bias and highest coverage, while BART MChains shows the lowest RMSE and interval length – it seems that ensemble methods based on a weak learner approach seem most suitable here. Finally, we notice that the Causal Forest performs better for GATEs estimation than for ATE estimation. When estimating ITEs, we find that, overall, BART MChains outperforms all methods, followed by DR MOM Boosting, DR MOM BART, Causal Forest and DR MOM Random Forest.

References

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

Athey, S., Imbens, G. W., and Wager, S. (2018). Approximate residual balancing: debiased inference of average treatment effects in high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(4):597–623.

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

Baiardi, A. and Naghi, A. A. (2021). The value added of machine learning to causal inference: Evidence from revisited studies. *arXiv preprint arXiv:2101.00878*.

Bertrand, M., Crépon, B., Marguerie, A., and Premand, P. (2017). Contemporaneous and post-program impacts of a public works program: Evidence from côte d'ivoire. Working Paper.

Carvalho, C., Feller, A., Murray, J., Woody, S., and Yeager, D. (2019). Assessing treatment effect variation in observational studies: Results from a data challenge. *Observational Studies*, 5(2):21–35.

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

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018a). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68.

Chernozhukov, V., Demirer, M., Duflo, E., and Fernandez-Val, I. (2018b). Generic machine learning inference on heterogenous treatment effects in randomized experiments. Working Paper, 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.

Colangelo, K. and Lee, Y.-Y. (2020). Double debiased machine learning nonparametric inference with continuous treatments. *arXiv preprint arXiv:2004.03036*.

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

Davis, J. M. and Heller, S. B. (2017). Rethinking the benefits of youth employment programs: The heterogeneous effects of summer jobs. *Review of Economics and Statistics*, pages 1–47.

Deryugina, T., Heutel, G., Miller, N. H., Molitor, D., and Reif, J. (2019). The mortality and medical costs of air pollution: Evidence from changes in wind direction. *American Economic Review*, 109(12):4178–4219.

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.

Farrell, M. H., Liang, T., and Misra, S. (2021). Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213.

Friedberg, R., Tibshirani, J., Athey, S., and Wager, S. (2020). Local linear forests. *Journal of Computational and Graphical Statistics*, pages 1–15.

Hahn, P. R., Dorie, V., and Murray, J. S. (2019). Atlantic causal inference conference (acic) data analysis challenge 2017. *arXiv preprint arXiv:1905.09515*.

Hill, J. and 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*, pages 1386–1420.

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

Jacob, D. (2021). Cate meets ml-the conditional average treatment effect and machine learning. *arXiv preprint arXiv:2104.09935*.

Knaus, M., Lechner, M., and Strittmatter, A. (2017). Heterogeneous employment effects of job search programmes: A machine learning approach.

Knaus, M. C., Lechner, M., and Strittmatter, A. (2018). Machine learning estimation of heterogeneous causal effects: Empirical monte carlo evidence. *The Econometrics Journal*.

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

McConnell, K. J. and Lindner, S. (2019). Estimating treatment effects with machine learning. *Health services research*, 54(6):1273–1282.

Neyman, J. (1959). Optimal asymptotic tests of composite hypotheses. *Probability and* *statsitics*, pages 213–234.

Neyman, J. (1979). C (α) tests and their use. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 1–21.

Niswander, K. R. and Gordon, M. (1972). *The women and their pregnancies: the Collaborative Perinatal Study of the National Institute of Neurological Diseases and Stroke*. National Institute of Health.

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

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

Semenova, V., Goldman, M., Chernozhukov, V., and Taddy, M. (2018). Orthogonal machine learning for demand estimation: High dimensional causal inference in dynamic panels. *arXiv preprint arXiv:1712.09988*.

Strittmatter, A. (2019). What is the value added by using causal machine learning methods in a welfare experiment evaluation? Working Paper.

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

Wendling, T., Jung, K., Callahan, A., Schuler, A., Shah, N., and Gallego, B. (2018). Comparing methods for estimation of heterogeneous treatment effects using observational data from health care databases. *Statistics in medicine*, 37(23):3309–3324.

A Supplementary Material to: *”Finite Sample Evaluation of Causal Machine Learning Methods: Guidelines for the Applied Researcher”*

A.1 Control variables selected from the Collaborative Perinatal Project

Table 5: Overview of control variables

variable	**description**	**variable**	**description**
x_1	mom_age	x_{30}	num_premes
x_2	mar_status	x_{31}	num_abortions
x_3	mom_cigs_per_day	x_{32}	num_prior_pregs
x_4	mom_years_smoked	x_{33}	num_stillbirths
x_5	mom_height	x_{34}	bayley_mental
x_6	mom_weight_prior	x_{35}	bayley_motor
x_7	mom_num_cardio_cond	x_{36}	placental_weight
x_8	mom_num_pulm_cond	x_{37}	cord_length
x_9	mom_num_hema_cond	x_{38}	sex
x_{10}	mom_num_endocrine_cond	x_{39}	apgar_1m_total
x_{11}	mom_num_veneral_cond	x_{40}	apgar_5m_total
x_{12}	mom_num_urin_cond	x_{41}	bottle_feed_days
x_{13}	mom_num_gyne_cond	x_{42}	breast_feed_days
x_{14}	mom_num_neur_cond	x_{43}	child_bilirubin
x_{15}	mom_num_obst_compl	x_{44}	child_hematocrit
x_{16}	mom_num_infect_dis	x_{45}	child_hemoglobin
x_{17}	mom_work_status	x_{46}	child_num_neur_abn
x_{18}	mom_years_educ	x_{47}	child_num_cns_cond
x_{19}	family_income	x_{48}	child_num_muscoskel
x_{20}	housing_density	x_{49}	child_num_resp_abn
x_{21}	mom_birth_place	x_{50}	child_num_cardio_abn
x_{22}	consanguinity	x_{51}	child_num_liver_abn
x_{23}	socio_eco	x_{52}	child_num_hemo_cond
x_{24}	mom_race	x_{53}	child_num_infect
x_{25}	age_menarche	x_{54}	child_num_synd
x_{26}	dias_blood_pres	x_{55}	child_num_endo_dis
x_{27}	mom_weight_birth	x_{56}	child_num_proc
x_{28}	dad_age	x_{57}	head_size_1yr
x_{29}	dad_years_educ	x_{58}	gest_delivery

Notes: The table gives an overview of the 58 control variables selected from the Collaborative Perinatal Project to capture confounding effects.

A.2 Overview of simulation scenarios in Dorie et al. (2019)

Table 6: Overview of all 77 simulation scenarios

Scen.	Treatment Model	Percent 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

Notes: The table shows the 77 simulations scenarios from the causal inference data competition by Dorie et al. (2019).

A.3 Additional results: overall performance

Table 7: Overall performance: ATE, averaged across all 77 simulation scenarios

	BART MC	Causal Forest	DML BART	DML Boost	DML Lasso	DML Nnet	DML RF	DML Trees	GML BART	GML Boost	GML Lasso	GML Nnet	GML RF	GML 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

Notes: The table displays the bias, RMSE, coverage rates, and interval lengths for the ATEs, where the results are averaged across all $k = 77$.

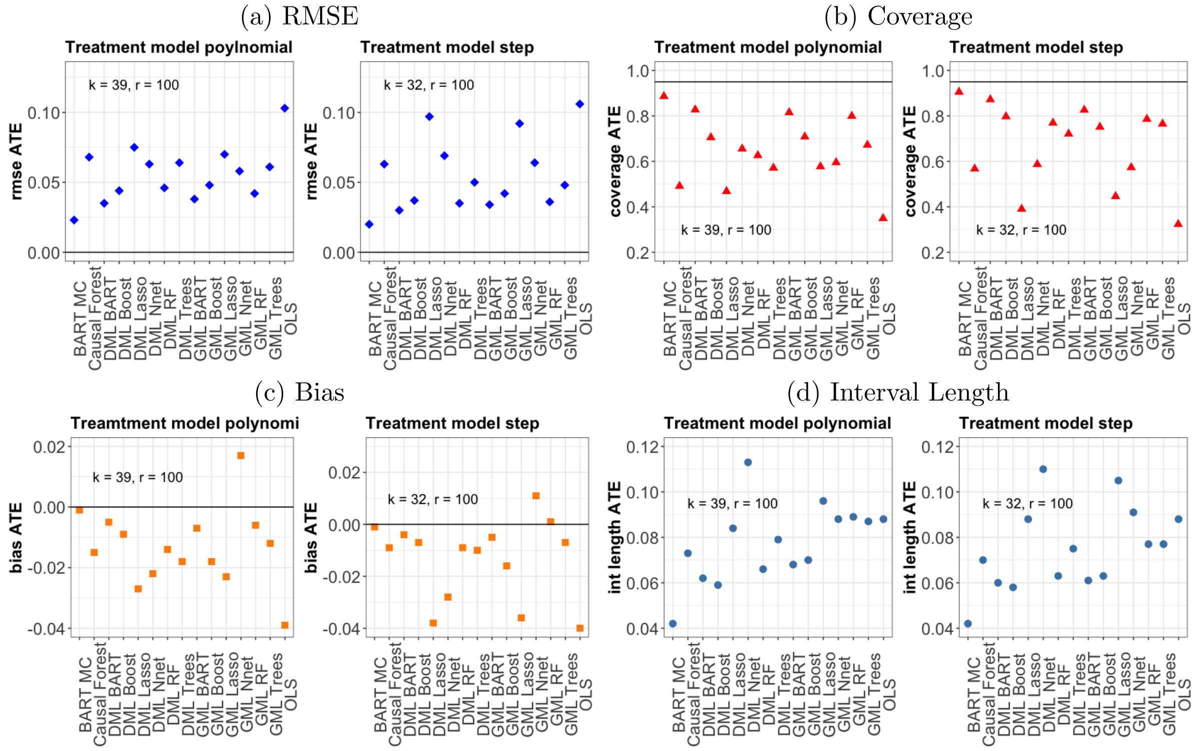
Table 8: Overall performance: GATEs, averaged across all 77 simulation scenarios

	BART MC		Causal Forest		GML BART		GML Boost		GML Lasso		GML Nnet		GML RF		GML 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

Notes: The table displays the bias, RMSE, coverage rates, and interval lengths for the GATEs, where the results are averaged across all $k = 77$.

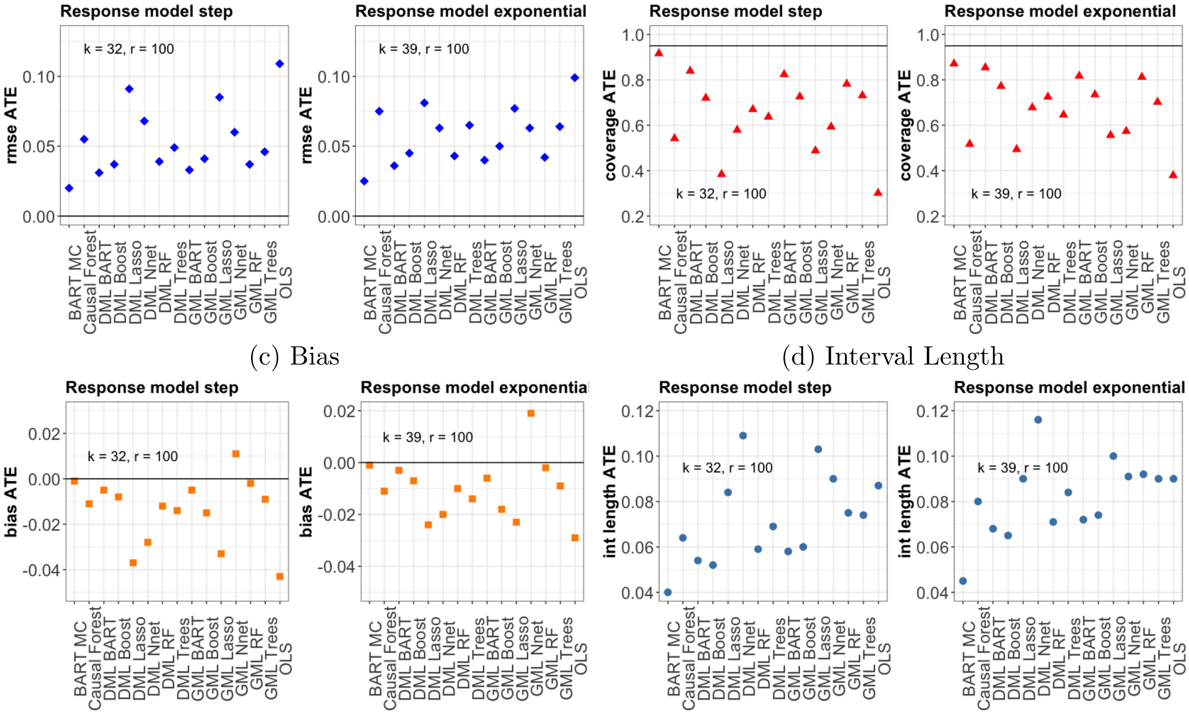
A.4 Additional results: analysis by criteria

Figure 5: Treatment model: polynomial and step, ATE



Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the ATE, where the results are averaged across all $k = 39$ scenarios with a polynomial treatment model versus all $k = 32$ scenarios with a step treatment model. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 6: Response model: step and exponential, ATE



(a) RMSE

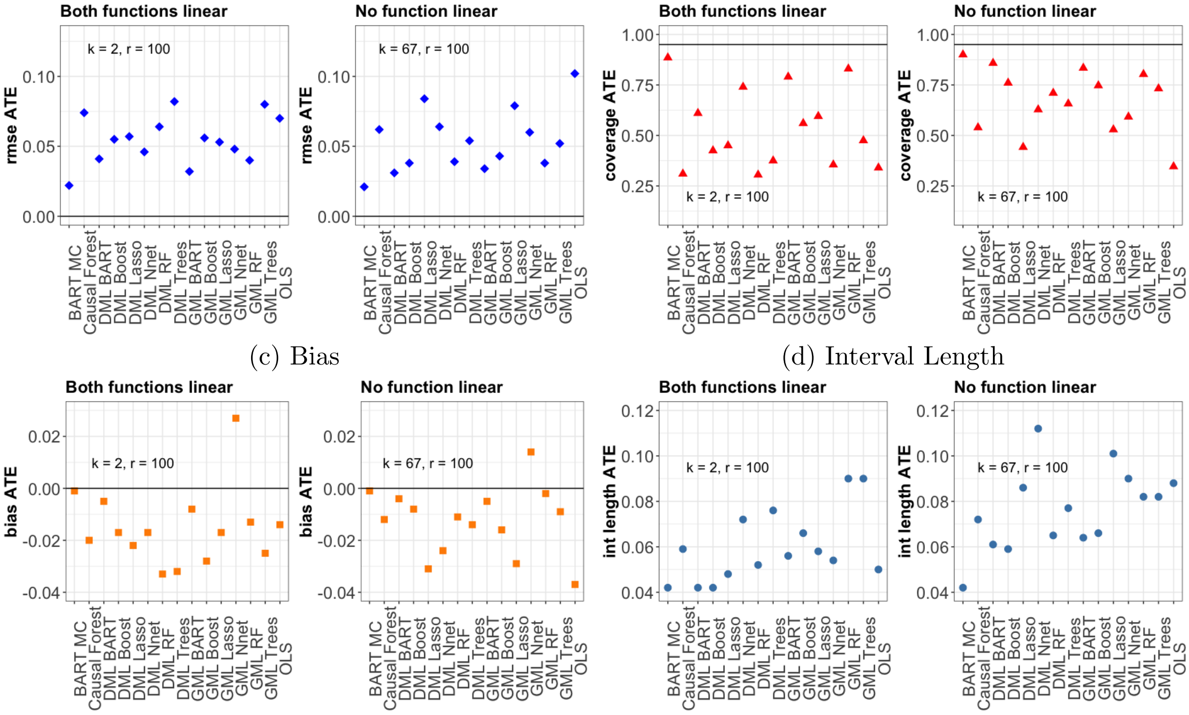
(b) Coverage

(c) Bias

(d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the ATE, where the results are averaged across all $k = 32$ scenarios with step response model versus all $k = 39$ scenarios with exponential response model. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 7: Treatment and Response model: linear and non-linear, ATE



(a) RMSE

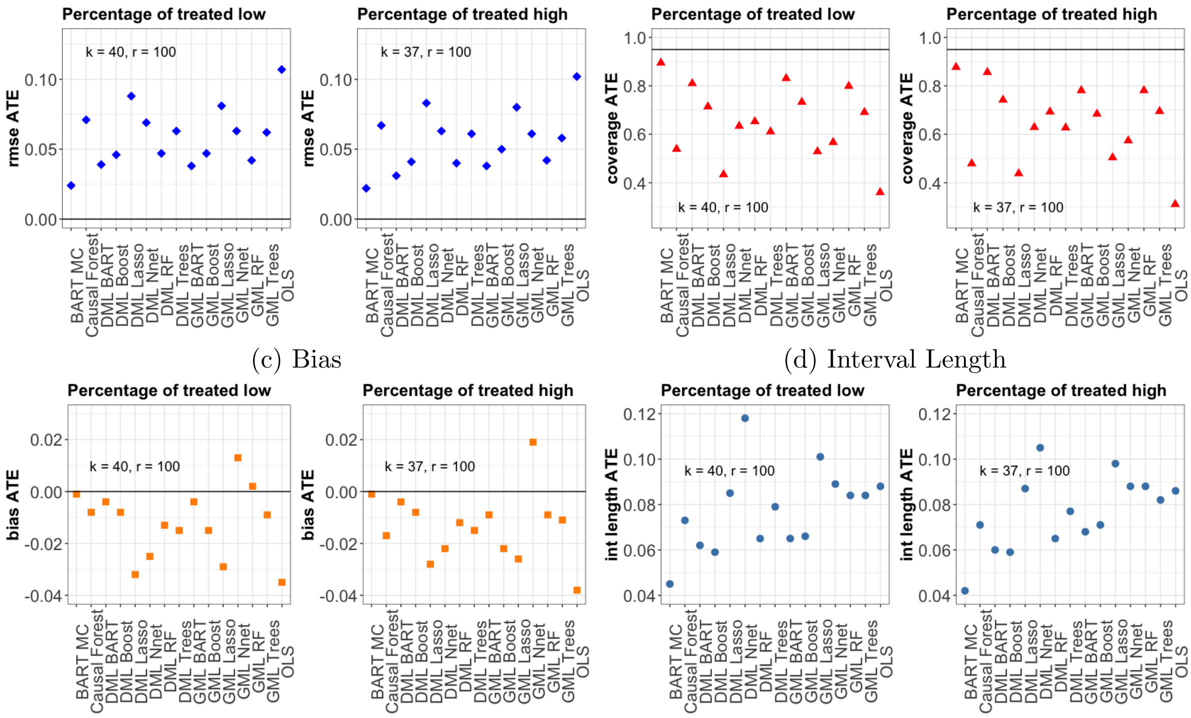
(b) Coverage

(c) Bias

(d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the ATE, where the results are averaged across all $k = 2$ scenarios where both models are linear versus all $k = 67$ scenarios where both models are non-linear. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 8: Percentage of treated: low and high, ATE



(a) RMSE

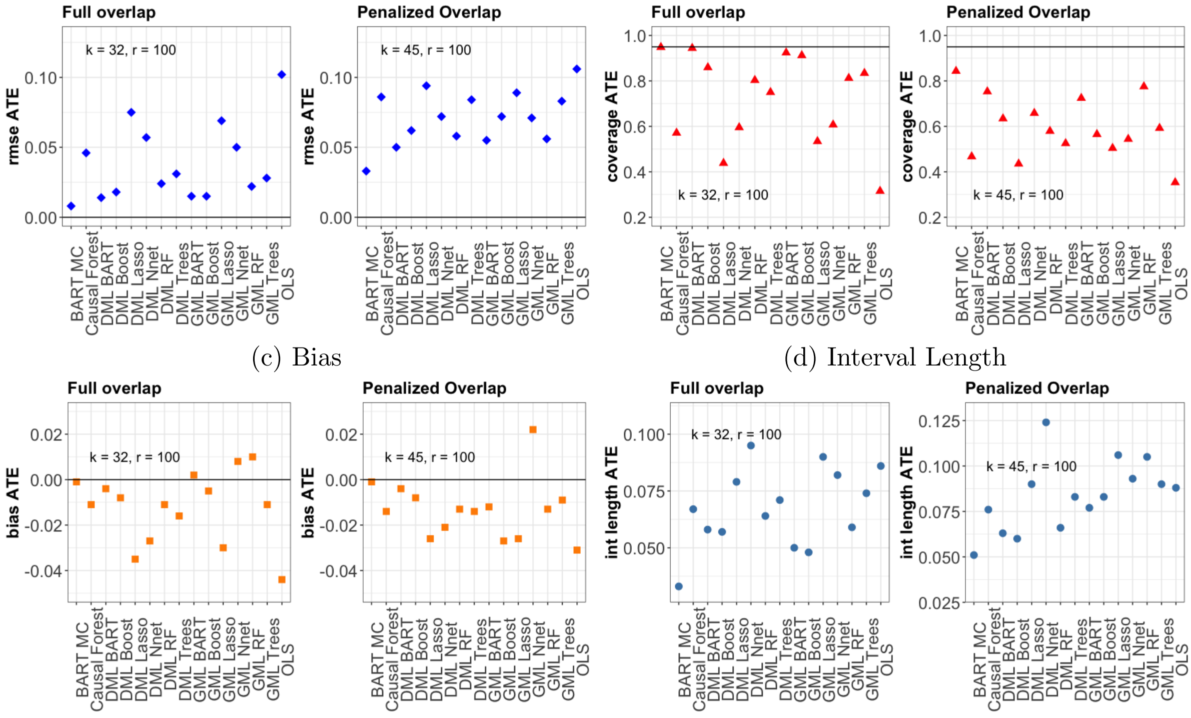
(b) Coverage

(c) Bias

(d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the ATE, where the results are averaged across all $k = 40$ scenarios with a low percentage of treated versus all $k = 37$ scenarios with a high percentage of treated. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 9: Overlap: full and penalized, ATE



(a) RMSE

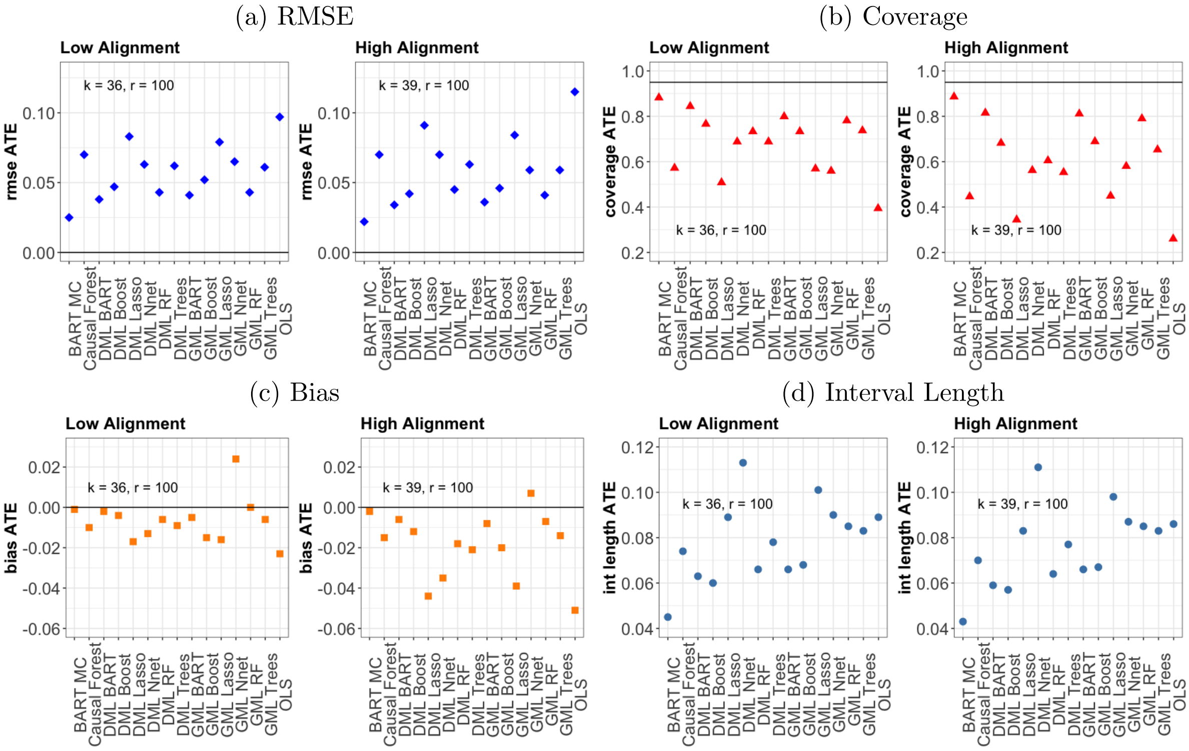
(b) Coverage

(c) Bias

(d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the ATE, where the results are averaged across all $k = 32$ scenarios with full overlap versus all $k = 45$ scenarios with penalized overlap. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 10: Alignment: low and high, ATE


(a) RMSE

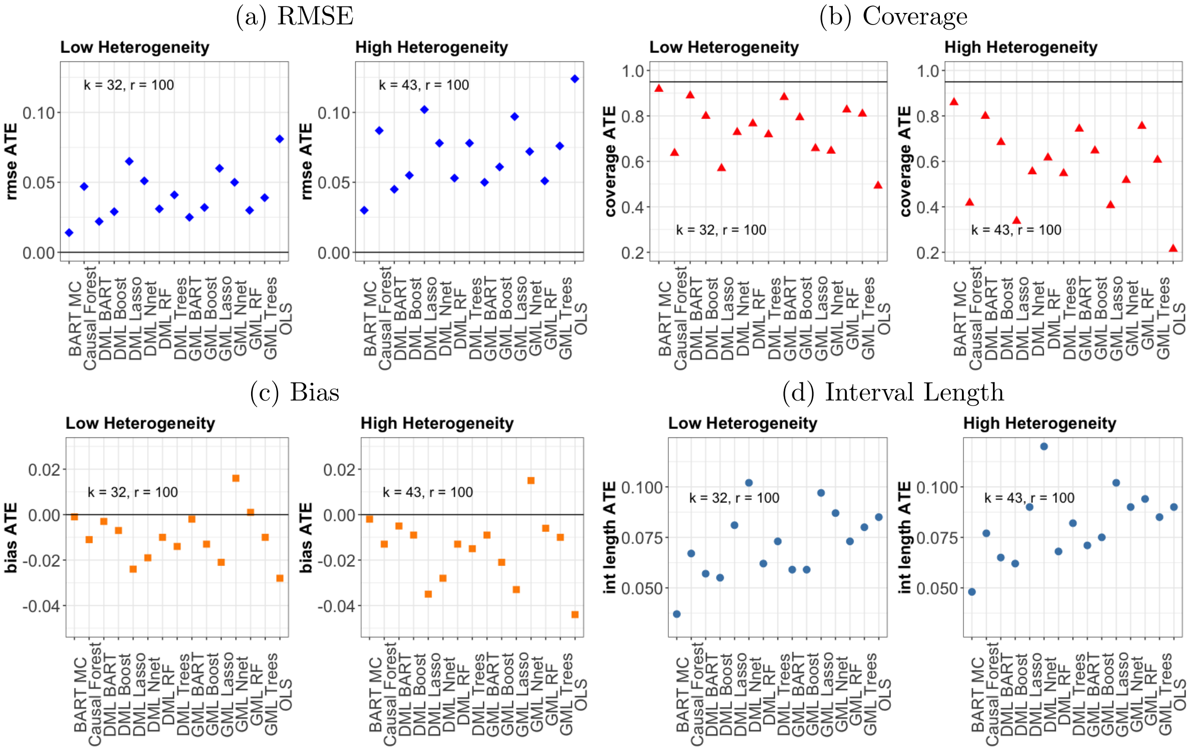
(b) Coverage

(c) Bias

(d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the ATE, where the results are averaged across all $k = 36$ scenarios with low alignment versus all $k = 39$ scenarios with high alignment. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 11: Heterogeneity: low and high, ATE


(a) RMSE

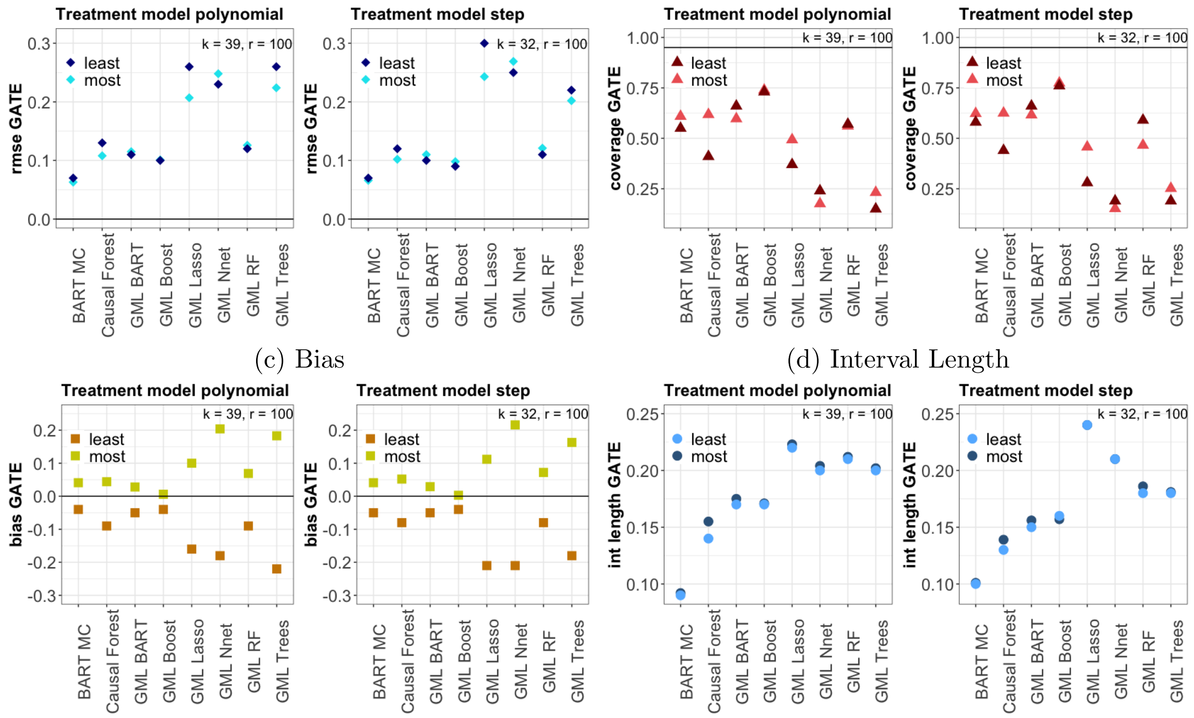
(b) Coverage

(c) Bias

(d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the ATE, where the results are averaged across all $k = 32$ scenarios with low heterogeneity versus all $k = 43$ scenarios with high heterogeneity. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 12: Treatment model: polynomial and step, GATE

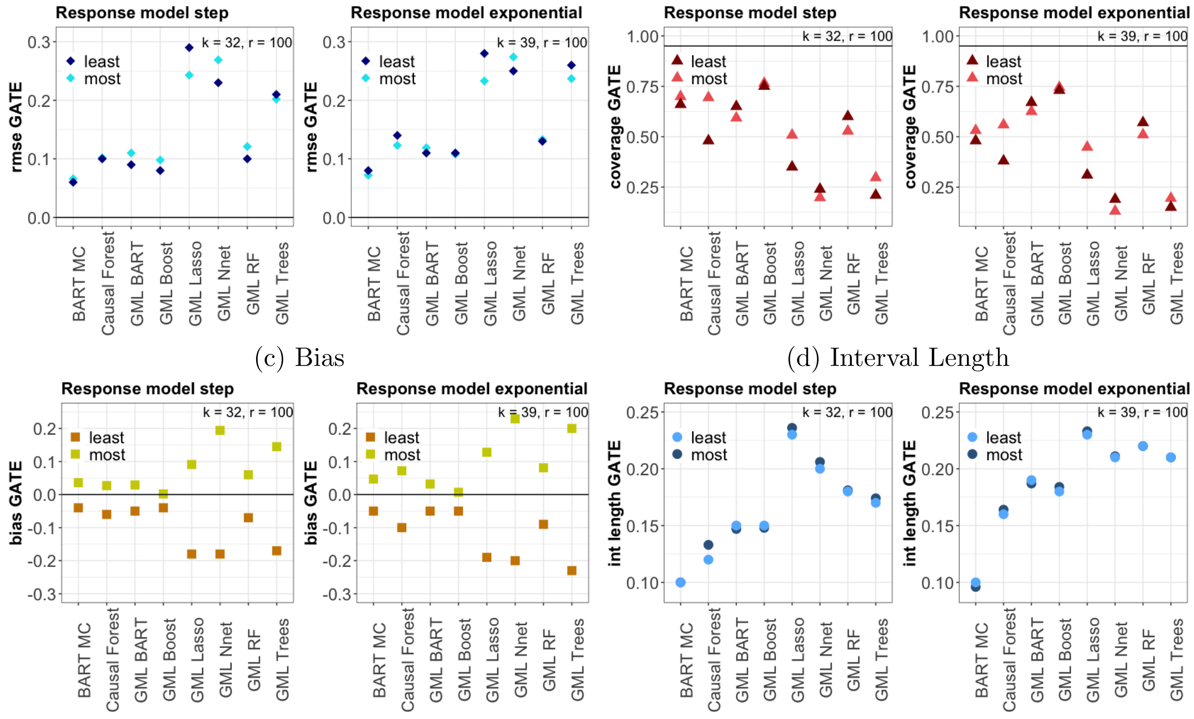


(a) RMSE (b) Coverage

(c) Bias (d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the GATEs, where the results are averaged across all $k = 39$ scenarios with a polynomial treatment model versus all $k = 32$ scenarios with a step treatment model. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 13: Response model: step and exponential, GATE

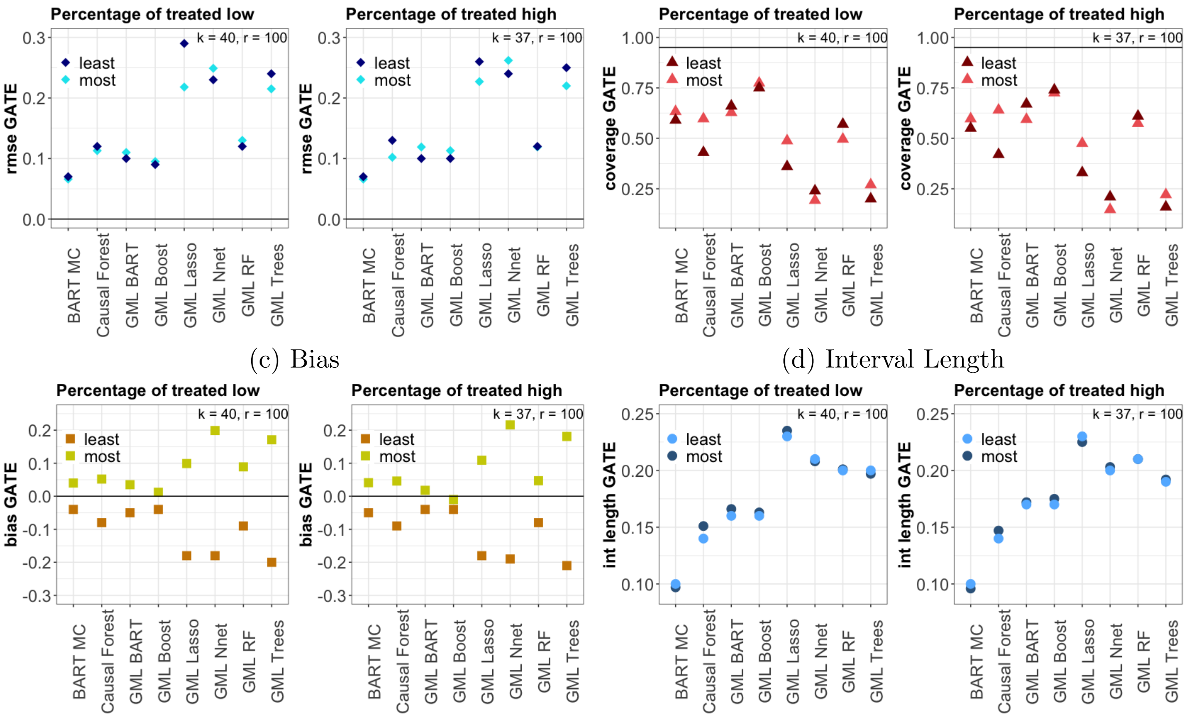


(a) RMSE (b) Coverage

(c) Bias (d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the GATEs, where the results are averaged across all $k = 32$ scenarios with full overlap versus all $k = 45$ scenarios with penalized overlap. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 14: Percentage of treated: low and high, GATE

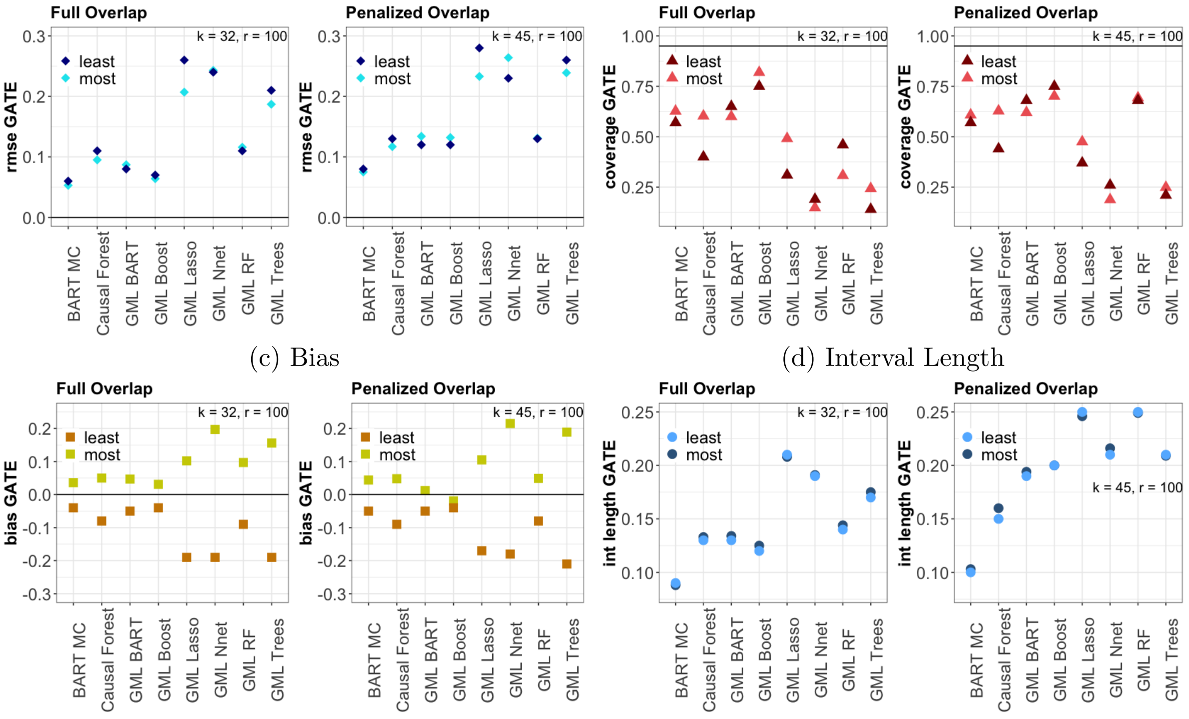


(a) RMSE (b) Coverage

(c) Bias (d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the GATEs, where the results are averaged across all $k = 40$ scenarios with a low percentage of treated versus all $k = 37$ scenarios with a high percentage of treated. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 15: Overlap: full and penalized, GATE

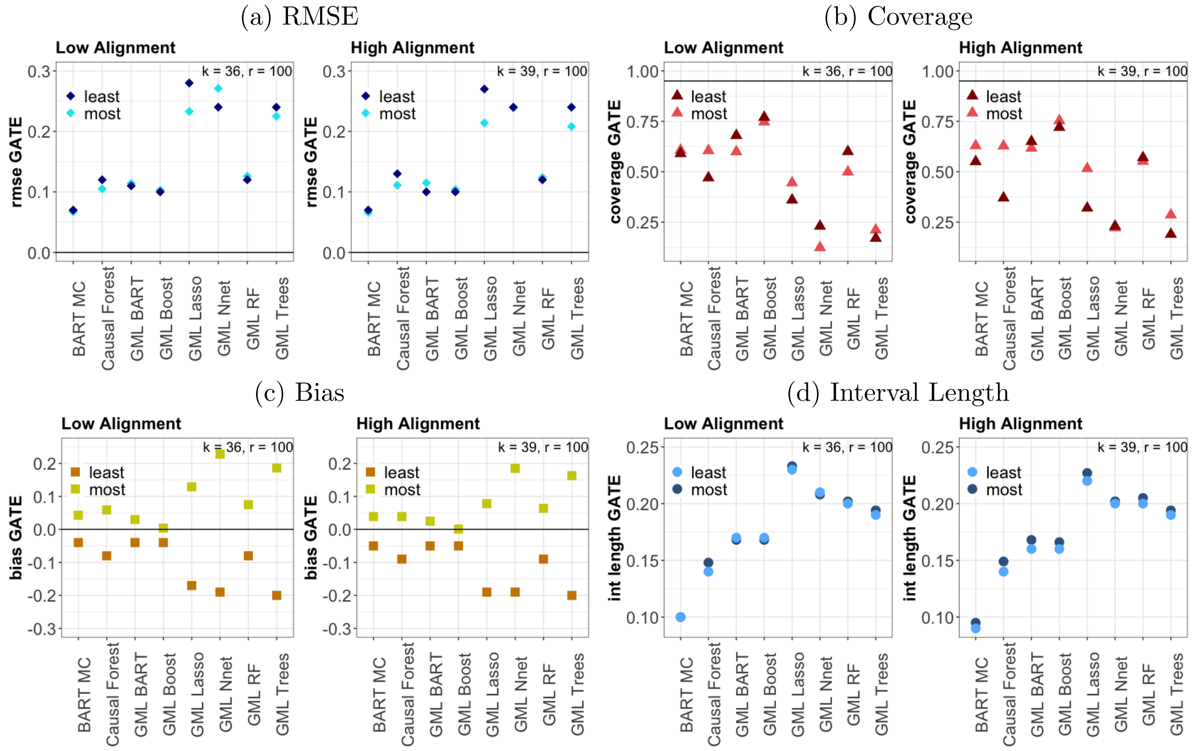


(a) RMSE (b) Coverage

(c) Bias (d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the GATEs, where the results are averaged across all $k = 32$ scenarios with full overlap versus all $k = 45$ scenarios with penalized overlap. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 16: Alignment: low and high, GATE


(a) RMSE

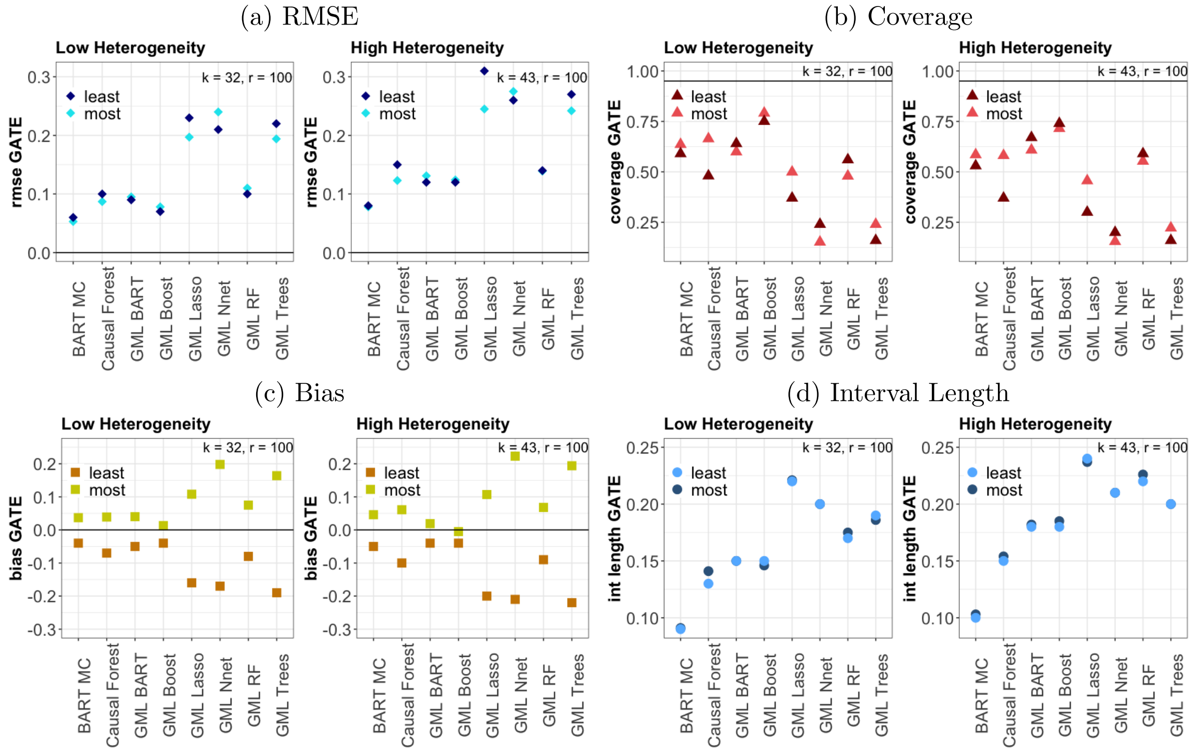
(b) Coverage

(c) Bias

(d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the GATEs, where the results are averaged across all $k = 36$ scenarios with low alignment versus all $k = 39$ scenarios with high alignment. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 17: Heterogeneity: low and high, GATE


(a) RMSE

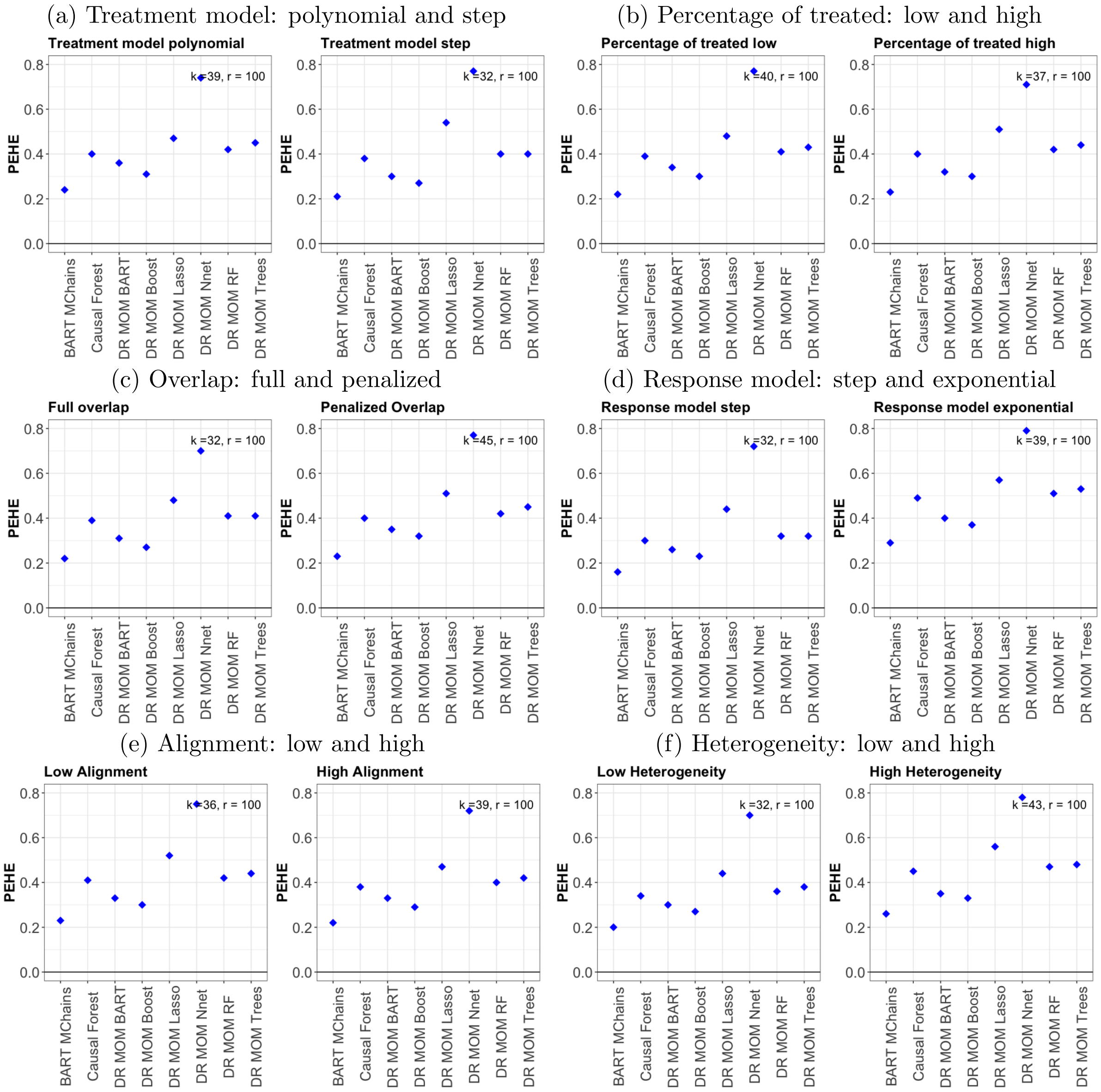
(b) Coverage

(c) Bias

(d) Interval Length

Note: The figure displays the bias, RMSE, coverage rates, and interval lengths for the GATEs, where the results are averaged across all $k = 32$ scenarios with low heterogeneity versus all $k = 43$ scenarios with high heterogeneity. The number of replications per simulation scenario is $r = 100$. The horizontal lines reflect the desired theoretical values for the bias, RMSE and coverage rate.

Figure 18: PEHE by Different Criteria, ITE



(a) Treatment model: polynomial and step

(b) Percentage of treated: low and high

(c) Overlap: full and penalized

(d) Response model: step and exponential

(e) Alignment: low and high

(f) Heterogeneity: low and high

Note: This figure displays the Precision in Estimating Heterogeneous Effects (PEHE) for the ITEs, by different criteria. The results are averaged across all k simulation scenarios. The number of replications per simulation scenario is $r = 100$.

A.5 Tuning parameters and Sensitivity Analysis

Table 9: Tuning parameters of the ML algorithms in the main analysis

Lasso (cv.glmnet)

λ**, regularization parameter**, controls the Lasso penalty. If λ increases the penalty shrinks more variables in the model fit which increases the bias but decreases the variance. It is chosen by minimizing the cross-validation error with 10-folds within the cv.glmnet package. In the Lasso we include polynomial terms up to the third order.

Trees (rpart)

cp**, complexity parameter**, controls the size of the decision tree by evaluating the complexity of the tree versus the goodness of fit to the data to avoid overfitting. The parameter is chosen by 10-fold cross validation within the rpart package.
minsplit, the minimum number of observations required in a node to consider a split in the tree. It is set to the default value of 20.
minbucket, the minimum size of the terminal nodes. It is set by default to the rounded value of minsplit/3.
maxdepth, the maximum depth of the final tree. It is set to the default value of 30.

Boosting (gbm)

n.trees, the number of trees that are sequentially grown within the Boosting algorithm. In the main analysis (Sections 4 and 5) 1000 trees are chosen. For the sensitivity analysis (Section 6) we also consider other values as specified in Table 11.
interaction.depth, specifies the maximum depth of each tree and it is set to 2 in the main analysis. In the sensitivity analysis we also consider other values as specified in Table 11.
n.minobsinnode, specifies the minimum amount of observations in the terminal node and it is set to 1 in the main analysis. For the sensitivity analysis we also consider the values specified in Table 11.
shrinkage, shrinkage parameter (also known as learning rate), controls the rate at which the boosting algorithm learns and guarantees that the update rule of each sequential tree leads to an overall improvement. The parameter is set by default to 0.1.
bag.fraction, the fraction of randomly selected training observations proposed for the next tree build, introducing randomness in the model fit. The parameter is set by default to 0.5.

Random Forest (randomForest)

ntrees, number of decorrelated trees to grow. In the main analysis 250 trees are chosen. For the sensitivity analysis we also consider other values as specified in Table 12.
mtry, number of variables randomly sampled as candidates at each split. The default values of $\frac{P}{3}$ for regression problems (response model) and $\sqrt{P}$ for classification problems (treatment model) are chosen, where P is the number of control variables. For the sensitivity analysis we also consider other values as specified in Table 12.
nodesize, the minimum size of the terminal nodes in each tree. It is set to the default values of 5 for regression problems (response model) and 1 for classification problems (treatment model).

Neural Net (nnet)

size, the number of neurons in the hidden layer. It is set to 2 in the main analysis. For the sensitivity analysis we also consider other values as specified in Table 13.
number hidden layers, it is set to the basic value of 1, as proposed in the implementation of the DML method.

decay, regularization term to avoid overfitting. If the decay increases, the bias on the training data increases but the variance on the test data decreases. It is set to 0.1 in the main analysis. For the sensitivity analysis we also consider other values as specified in Table 13.
maxit, the maximum number of iterations within the Neural Net. It is set to 1000.

Causal Forest (grf)

num.trees, the number of trees grown in the forest. It is set to the default value of 2000.
mtry, the number of variables randomly sampled as candidates at each split. It is tuned within the grf package.
min.node.size, minimum number of observations in each tree's leaf. It is set to the default value of 5.
sample.fraction, fraction of data used for the sub-sample (see Section 3.3). It is tuned within the grf package.
honesty.fraction, fraction of data used as training sample (see Section 3.3). It is tuned within the grf package.
alpha, regularizes the maximum imbalance possible at each split. It is tuned within the grf package.

BART (BART)

ntrees, the number of trees in the sum which are grown in the BART algorithm. It is set to the default of 200.
ndpost, the number of posterior draws returned (iterations in the MCMC). It is set to the default of 1000.
nskip, the number of burn-ins in the MCMC iterations. It is set to the default of 100.

BART MC (bartCause)

n.chains, number of combined chains. It is set to the default of 10.
commonSup.rule, a rule defining which data points to exclude in case of poor overlap. Rule ”sd” is chosen, which excludes observations whose predicted counterfactual standard deviation is extreme.

Notes: The table gives information on the tuning parameters used for the Machine Learning methods in the main analysis. The name in parentheses next to the ML methods denotes the corresponding R-package used.

Table 10: Overview of fixed and tuned parameters

Parameter	(1) Fixed	(2) Tuned within package	(3) Tuned with *caret* package
Lasso			
λ, regularization parameter		✓	
Trees			
***cp*, complexity parameter**		✓	
minsplit	✓		
minbucket	✓		
maxdepth	✓		
Boosting			
n.trees	✓		✓
interaction.depth	✓		✓
n.minobsinnode	✓		✓
shrinkage	✓		
bag.fraction	✓		
Random Forest			
ntrees	✓		✓
mtry	✓		✓
nodesize	✓		
Neural Net			
size	✓		✓
number hidden layers	✓		
decay	✓		✓
maxit	✓		
Causal Forest			
num.trees	✓		
mtry		✓	
min.node.size	✓		
sample.fraction		✓	
honesty.fraction		✓	
alpha		✓	
BART			
ntrees	✓		
ndpost	✓		
nskip	✓		
BART MC			
n.chains	✓		

Notes: The table gives an overview of fixed and tuned parameters. Columns (1) and (2) are the parameter setting in Sections 4 and 5. Column (3) shows which parameters have been additionally varied in the sensitivity analysis in Section 6.

Table 11: Tuning parameter combinations, Boosting

Parameter Combination	1	2	3	4	5	6	7	8	9	10	11	12
n.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
n.minobsinnode	1	1	1	1	1	1	5	5	5	5	5	5

Notes: The table specifies the different parameter combinations in the sensitivity analysis. Each column represents a possible combination of the numbers of trees, the interaction depth and the minimum number of observations in the terminal node, as inputs for the Boosting method. The parameter values are manually predefined.

Table 12: Tuning parameter combinations, Random Forest

Parameter Combination	1	2	3
ntrees	500	500	500
mtry	22	27	32

The table specifies the different parameter combinations in the sensitivity analysis. Each column represents a possible combination of the numbers of trees and the numbers of variables randomly sampled to be used for splits, as inputs for the Random Forest method. The parameter values are manually predefined.

Table 13: Tuning parameter combinations, Neural Net

Parameter Combination	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

The table specifies the different parameter combinations in the sensitivity analysis. Each column represents a possible combination of the size and decay parameter as inputs for the Neural Net method. The parameter values are manually predefined.

Table 14: Sensitivity Check: ATE performance analysis by different criteria/data features

Scenario	DML Boost	DML Nnet	DML RF	GML Boost	GML Nnet	GML RF
			Panel A: Bias			
27	-0.00	-0.01	-0.01	-0.00	0.02	0.01
55	-0.00	-0.01	-0.00	-0.00	0.02	0.01
40	-0.00	-0.01	-0.00	-0.00	0.03	0.01
21	-0.00	-0.01	-0.00	-0.01	0.03	0.00
31	-0.00	-0.01	-0.00	0.00	0.03	0.01
29	-0.01	-0.03	-0.02	-0.00	0.01	0.00
28	-0.01	-0.03	-0.01	-0.01	0.01	0.01
			Panel B: RMSE			
27	0.01	0.03	0.02	0.01	0.04	0.02
55	0.01	0.04	0.01	0.01	0.05	0.02
40	0.01	0.03	0.02	0.01	0.04	0.02
21	0.03	0.04	0.03	0.03	0.05	0.03
31	0.01	0.03	0.02	0.01	0.04	0.02
29	0.01	0.06	0.03	0.01	0.04	0.02
28	0.02	0.06	0.03	0.01	0.05	0.02
			Panel C: Coverage			
27	0.96	0.83	0.90	0.97	0.74	0.85
55	0.96	0.85	0.97	0.98	0.66	0.76
40	0.96	0.84	0.87	0.96	0.63	0.86
21	0.75	0.87	0.74	0.76	0.65	0.86
31	0.99	0.91	0.96	0.93	0.56	0.86
29	0.82	0.59	0.65	0.92	0.62	0.90
28	0.93	0.54	0.71	0.91	0.54	0.83
			Panel D: Interval Length			
27	0.05	0.10	0.05	0.04	0.09	0.05
55	0.05	0.09	0.05	0.04	0.08	0.05
40	0.05	0.09	0.05	0.04	0.08	0.05
21	0.05	0.12	0.05	0.06	0.10	0.07
31	0.06	0.10	0.07	0.05	0.08	0.06
29	0.04	0.09	0.06	0.04	0.07	0.06
28	0.06	0.10	0.07	0.04	0.08	0.06

Notes: The table reports the results on the ATE sensitivity checks of the ML input parameters. The input parameters have been chosen by the best combination resulting from the values defined in Tables 11-13. 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 of the $r = 100$ simulation replications.

Table 15: Sensitivity Check: GATE performance analysis by different criteria/data features

Scenario	GML Boost		GML Nnet		GML RF	
	most affect.	least affect.	most affect.	least affect.	most affect.	least affect.
			Panel A: Bias			
28	0.03	-0.04	0.22	-0.20	0.09	-0.08
56	0.03	-0.03	0.21	-0.22	0.09	-0.06
41	0.04	-0.04	0.22	-0.20	0.09	-0.07
22	0.01	-0.03	0.26	-0.20	0.08	-0.08
32	0.04	-0.05	0.26	-0.23	0.11	-0.11
30	0.04	-0.05	0.20	-0.21	0.09	-0.10
27	0.03	-0.04	0.20	-0.16	0.08	-0.06
			Panel B: RMSE			
28	0.06	0.07	0.28	0.24	0.11	0.10
56	0.07	0.07	0.28	0.29	0.10	0.08
41	0.07	0.07	0.26	0.26	0.11	0.10
22	0.14	0.10	0.31	0.26	0.17	0.12
32	0.06	0.08	0.30	0.26	0.13	0.13
30	0.10	0.08	0.28	0.25	0.13	0.12
27	0.04	0.05	0.24	0.20	0.09	0.07
			Panel C: Coverage			
28	0.84	0.76	0.14	0.16	0.48	0.51
56	0.78	0.82	0.20	0.24	0.30	0.54
41	0.75	0.74	0.14	0.13	0.43	0.52
22	0.63	0.76	0.21	0.26	0.66	0.75
32	0.77	0.73	0.12	0.12	0.35	0.42
30	0.74	0.69	0.22	0.12	0.40	0.35
27	0.72	0.71	0.13	0.26	0.40	0.64
			Panel D: Interval Length			
28	0.12	0.12	0.20	0.19	0.15	0.14
56	0.11	0.11	0.21	0.21	0.13	0.13
41	0.12	0.12	0.19	0.19	0.14	0.14
22	0.18	0.18	0.25	0.24	0.27	0.25
32	0.14	0.14	0.20	0.20	0.16	0.16
30	0.12	0.12	0.19	0.18	0.15	0.15
27	0.11	0.11	0.20	0.19	0.13	0.13

Notes: The table reports the results on the GATEs sensitivity checks of the ML input parameters. The input parameters have been chosen by the best combination resulting from the values defined in Tables 11-13. 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 of the $r=100$ simulation replications.