Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis
BSc^2 Econometrics and Econonomics

A comparison and sensitivity analysis of double machine learning methods for treatment effect inference in high-dimensional partially linear settings

Maurizio Raina (497726)

Supervisor:	Sven Koobs
Second assessor:	Jeffrey Durieux
Date final version:	1st July 2024

Abstract

This thesis investigates the comparative performance of double machine learning (DML) and LASSO-based methods for treatment effect inference in high-dimensional, partially linear settings. Given the increasing prevalence of high-dimensional data and the inadequacy of traditional estimation methods like ordinary least squares, advanced machine learning techniques are increasingly used for unbiased and efficient treatment effect estimation. The primary focus is on replicating the results of Belloni et al., 2011 and comparing them with the DML framework introduced by (Chernozhukov et al., 2018). Monte Carlo simulations are used to evaluate the methods across various scenarios. The results of this sensitivity analysis demonstrate that DML methods generally outperform Lasso-based methods in terms of bias and efficiency, providing more robust estimates of treatment effects. This research aims to equip impact evaluation practitioners with practical guidelines on selecting the appropriate model based on data characteristics and assumptions. The findings contribute to the existing literature by offering a comprehensive comparison of these advanced methods, highlighting their relative strengths and weaknesses in diverse data-generating processes.

Contents

1 Introduction **3**

2 Literature review **4**

2.1 Economics literature 4

2.2 Econometrics literature 6

3 Theoretical Framework: Machine Learning Methods **7**

3.1 Regression trees 7

3.1.1 Random Forest 7

3.1.2 Extreme Gradient Boosting 8

3.2 Least Absolute Shrinkage and Selection Operator 8

4 Data **9**

4.1 Monte Carlo simulations 9

4.1.1 Replication simulation 9

4.2 Extension simulations 10

4.2.1 simulation 1: default simulation 10

4.2.2 simulations 1.1-1.21: tuning number of simulations 11

4.2.3 simulations 2.1-2.10: Sensitivity to sample size 11

4.2.4 simulations 3.1-3.10: Sensitivity to noise covariates 12

4.2.5 simulations 4.1-4.10: Sensitivity to confounding strength 12

4.2.6 simulations 5.1-5.10: sensitivity to numbers of confounders s 12

4.2.7 simulations 6.1-6.10, and 7.1-7.10: Sensitivity to number of variables that only affect the outcome and the treatment 13

4.2.8 simulations 8.1-8.10: sensitivity to unobserved confounders 13

4.2.9 simulations 9: including interaction and squared terms 14

4.2.10 simulations 10: hyper-parameter tuning of RF 14

5 Methodology **15**
5.1 Post-Lasso estimation 15
5.2 Double Machine Learning Methods 15
5.3 Other Extension Benchmarks 16

6 Results **16**
6.1 Replication results 16
6.2 Extension simulations results 17
6.2.1 simulation 1.1-1.21: tuning number of Monte-Carlo iterations 17
6.2.2 simulations 2.1-2.10: Sensitivity to sample size 19
6.2.3 simulations 3.1-3.10: Sensitivity to noise covariates 21
6.2.4 simulations 4.1-4.10: Sensitivity to confounding strength 22
6.2.5 simulations 5.1-5.10: sensitivity to numbers of confounders s 22
6.2.6 simulations 6.1-6.10: sensitivity to covariates that just affect the outcome 25
6.2.7 simulations 7.1-7.10: sensitivity to covariates that just affect the treatment 25
6.2.8 simulations 8.1-8.10: sensitivity to unobserved confounders 28
6.2.9 simulations 9: including interaction and squared terms 28
6.2.10 simulations 10: hyperparameter tuning of RF 33

7 Discussion and conclusion **34**

8 Acknowledgments **35**

A Programming code appendix **38**

B Supplementary figures appendix **39**

C Supplementary tables appendix **48**

1 Introduction

In the first decades of the 21st century, we are witnessing a revival of Machine Learning (ML) methods. This phenomenon is mainly due to the increasing amount of data collected and decreasing costs of computing power (Fradkov, 2020). The prevalence of high-dimensional data poses the challenge that not only the number of observations increases but also the number of covariates, making traditional methods, such as ordinary least squares (OLS), estimation unfeasible (Donoho et al., 2000). Consequently, new ML methodologies have been developed to address these challenges.

So far, ML methods have been used primarily in forecasting. The focus of this thesis lies on comparing the relative performances of newer ML methods in the context of inferring a partially linear treatment effect with a continuous treatment variable. By partially linear treatment effect, it means that the effect of treatment on outcome is modelled or simulated as linear, while other covariates can non-linearly influence the outcome and the treatment (for more details, see Section 4). The main methods compared are Lasso-based methods by Belloni et al., 2011, 2013 and double debiased machine learning (DML) methods introduced in Chernozhukov et al., 2018. Therefore, the main research question is as follows:

Research Question: How do replication Lasso-based and DML methods compare to one another regarding bias of treatment effect in partially linear high-dimensional settings?

There are several aspects to consider when answering the research question. Firstly, focusing on Belloni et al., 2011, 2013 methods, the goal is to replicate the methods and the results with the information provided by the authors:

Subquestion 1: Are the results on double-selection inference by Belloni et al. (2011, Section 6.2) replicable?

The output will be a table with mean bias, bias standard deviation, and implied 95% coverage rejection proportion analogous to that in Belloni et al. (2011, Section 6.2). These results are then compared with a more general DML framework. The best-performing replication method, double-selection, can be seen as a particular case of DML-Lasso inference where the Lasso penalties are those derived in Belloni et al. (2011, Appendix) (see Section 5). Thus, several additional DML extension models and benchmarks are then added to the simulation models and another ad-hoc simulation setup to evaluate the comparative performance of the competing models:

Subquestion 2: How do the results of the replication study compare in terms of bias performances to the DML methods?

Literature extensively reports that a newer set of ML methods for causal inference outperforms more traditional impact evaluation methods and competing ML algorithms in terms of bias, efficiency, and coverage (Belloni et al., 2013, 2014, 2016; Caron et al., 2022; Fuhr et al., 2024; McConnell & Lindner, 2019). However, these relatively recent methods are mainly used and understood only by academics. This thesis aims at impact evaluation practitioners tasked with causal inference who need to know what model is more appropriate for the dataset that is utilized. In turn, the results can be used to provide a more robust and unbiased impact evaluation inference in many fields where high-dimensional data is present, such as pharmaceutical trials, marketing, and policy evaluation. These developments allow us to assess the impact of

policies more accurately, helping to inform evidence-based interventions.

Subquestion 3: Can practical guidelines for impact evaluation practitioners be derived concerning which model is preferable under which scenario?

To answer *Subquestion 3*, an extensive sensitivity analysis of the data-generating process (DGP) of the performances is carried out. The main focus is investigating how the models' relative performance, in terms of producing an unbiased and well-behaved treatment effect estimate, changes as some key DGP parameters and assumptions are relaxed.

Subquestion 4: Sensitivity analysis: how is the relative performance of the models affected by changes in the data-generating process parameters?

This research compares two recently developed and high-performing suites of ML methods for causal inference in high-dimensional datasets. These methods have yet to be systematically evaluated against each other. Fuhr et al. (2024) compared DML methods, but under the simplifying orthogonal assumption of exogeneity that will be relaxed in this thesis, and not taking into account the Lasso methods from Belloni et al., 2011, 2013, 2014. It has practical relevance by providing guidelines to practitioners on which method to use based on data characteristics and assumptions. Existing literature has proposed these methods separately but lacks a comprehensive comparison, leaving practitioners uncertain about their relative strengths and weaknesses. This research will fill that gap by focusing on the performance of these methods in different data scenarios, which are currently comparatively poorly understood.

The main results of this study show that DML methods generally outperform LASSO-based methods in terms of bias and efficiency when estimating treatment effects in high-dimensional, partially linear settings. Specifically, the DML-RF and DML-LASSO methods provide more robust and unbiased estimates compared to traditional and LASSO-based methods from Belloni et al., 2011, 2013, which often suffer from higher biases and incorrect rejection rates.

The remainder of the thesis is structured as follows. In Section 2, this research is linked to academic literature, and critical concepts are defined. Section 3 introduces some theoretical background concerning ML methods. In Section 4, the Monte Carlo simulation setups for the sensitivity analysis are illustrated. Section 5 details the implementation of the LASSO and DML based methods compared in the thesis. The results of the replication study and extension are reported in Section **??**, and finally, Section 7 concludes with the discussion. The appendices at the end of the document contain instructions on using the replication file and additional tables and figures. 5

2 Literature review

2.1 Economics literature

The economic field of impact evaluation has extensively developed and assessed methods for estimating causal inference. In this context, estimating the treatment effect of a policy in an unbiased and econometrically sound manner, free from critiques of violated assumptions, is crucial. The most popular methods in impact evaluation literature revolve around searching for the perfect counterfactual to estimate the treatment effect. The counterfactual refers to the outcome if the treated population was left untreated. In probability notation, this idea is

expressed as: $\alpha_0 = \mathbb{E}(Y|T \neq 0, X) - \mathbb{E}(Y|T = 0, X)$, where Y denotes the outcomes, X the relevant characteristics, T the treatment, and α_0 the true treatment effect. Finding a valid counterfactual means estimating the hypothetical situation $\mathbb{E}(Y|T = 0, X)$. Following now is a summary of the methods most commonly used to look for a counterfactual (Gertler et al., 2016).

One of the most robust methods for finding a counterfactual is randomized assignment, meaning that individuals in the control group (non-treated) and treatment group are chosen randomly, resulting in two groups with individuals with approximately the same confounding characteristics. However, assigning policy treatments in a randomized manner in economic applications is often impractical or unethical. Instead, researchers often have to work with observational data, where some characteristics may be related to the propensity to receive treatment and the outcome; this phenomenon is known as confounding (see Figure 1). If randomized assignment does not lead to similar groups in terms of confounding variables, this results in omitted variable bias in the estimate of the treatment effect.

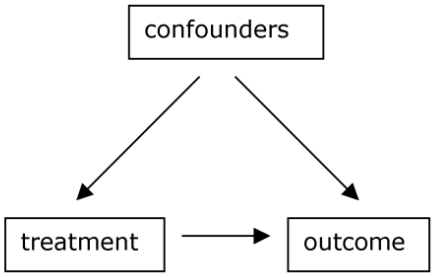

Figure 1: Confounding diagram. The directed arrows represent causality relationship among variables, where A $\rightarrow$ B means A is exogenous and explains B.

Another robust causal inference method used to overcome the challenge of potential omitted variable bias is the utilization of instrumental variables (IVs). As shown in Figure 2, an IV correlates with the treatment but is uncorrelated with the residuals and other (potentially unobserved) confounding variables, affecting the outcome only via its effect on the endogenous treatment due to the exclusion restriction. The typical estimation method in the linear context consists of two-stage least squares (2SLS). Let Z denote an instrument and T the endogenous treatment variable ($\mathbf{E}(T'\varepsilon) \neq 0$). In the first stage, the fit $X|Z$ is estimated. An IV is considered 'strong' if the instrument strongly correlates with the endogenous regressors. The second stage regresses the outcome Y on treatment T and the $X|Z$ fit to derive a treatment effect uncontaminated by omitted variable bias due to confounding. Despite desirable theoretical properties, finding valid and strong instruments in practice is often challenging. Staiger and Stock, 1994 show that if the instrument is weak, it complicates the identification of a valid counterfactual. The 2SLS approach can result in estimates biased towards the OLS estimate, both in small samples and asymptotically. Belloni et al. (2014) estimates the effects of abortions on crime rates, a topic previously approached in the literature via differences-in-differences estimation and IVs (Donohue & Levitt, 2001; Levitt, 1996). However, when considering models with more controls than observations, unintuitive instruments consisting of higher-order interaction terms have been found to be stronger than instruments derived solely by considering linear IVs and linear controls.

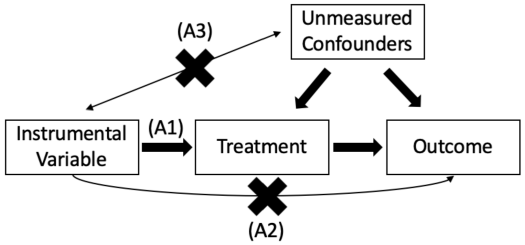

Figure 2: Diagram of assumptions required for a valid Instrumental Variable. The directed arrows represent causal relationships among variables. Adapted source: Johnson et al., 2021.

Other methods often used in practice to argue for a valid counterfactual and thus unbiased treatment effect include matching (e.g., sibling effects, synthetic control), regression discontinuity design, and difference-in-differences. All these methods rely on strong assumptions, making the internal validity of the results always questionable in real-data scenarios (Gertler et al., 2016). For example, valid difference-in-differences inference requires two assumptions to hold over time: *i.* constant treatment effect and *ii.* parallel trends. Recent efforts in the literature have focused on augmenting these impact evaluation methods to relax the strict assumptions they require for valid inference, such as allowing for dynamic treatment effects (Goodman-Bacon, 2021).

2.2 Econometrics literature

This thesis aims to contribute to a newer area of research, focusing on the increased availability of computing power and high-dimensional data, where the number of potential confounding variables p may be higher than observations n. Traditional methods based on OLS specifications often fail (rank condition violated). To address these challenges, researchers have developed methods for causal inference by adapting Machine Learning (ML) algorithms, which have shown strong performance in forecasting applications. The main challenge in developing these novel ML inference approaches is dealing with the regularization bias that a direct application of these ML estimation methods in high-dimensional datasets would entail. Regularization bias refers to the bias introduced into parameter estimates when regularization techniques are applied to prevent overfitting (Chernozhukov et al., 2018). Belloni et al. (2011, 2014) propose a three-step procedure to remove shrinking bias from the treatment effect estimator by assuming approximate sparsity. This procedure is called double selection and involves:

1. Running a least absolute shrinkage and selection operator (LASSO) to select control variables X that predict the outcome y.

2. Running another LASSO to select X that predicts the treatment variable D.

3. Estimating the treatment effect via LS on the variables selected in steps *i.* and *ii.*.

The LASSO penalty parameters λ are modified compared to the original Tibshirani (2018) ones to ensure robustness to heteroskedastic and non-Gaussian data generating processes (DGP) with desirable asymptotic properties (Belloni et al., 2012; Belloni et al., 2011, 2016; Tibshirani, 2018).

Another more generalized approach to applying ML methods (other than LASSO) in causal inference is double/debiased machine learning (DML) approaches. This suite of methods is based on the Frisch-Waugh-Lovell theorem of residual-on-residual regression to control for confounders via Neyman-orthogonalization (Frisch & Waugh, 1933; Lovell, 1963). Through sample splitting, the residuals can be obtained by two ML predictions (E[Y—X] and E[D—X]), where part of the sample is used to train the algorithm and the rest to predict, thus computing the residuals by prediction error. Sample splitting, formally known as "k-fold cross-validation", addresses regularisation bias and avoids overfitting. Additionally, averaging and correcting the standard errors (SE) to retrieve efficiency is performed. The process is repeated multiple times to mitigate the bias induced by the specific choice of splitting points (Chernozhukov et al., 2018; Fuhr et al., 2024; McConnell & Lindner, 2019). Many different ML models are possible for variable selection or regression-based ML forecasting. To limit the scope of this paper, just three ML algorithms to implement DML methods will be considered: LASSO and the other two tree-based algorithms: random forest (RF) and extreme gradient boosting (XGB).

3 Theoretical Framework: Machine Learning Methods

3.1 Regression trees

Regression trees (RT) are a fundamental machine learning method utilized for regression tasks. They operate by recursively partitioning the space of the covariates into smaller, more manageable subsets. This partitioning is carried out through a series of splits, each defined by a decision rule on one of the covariates. The result is a tree structure where each node represents a split on a covariate, and the leaves represent subsets of the data with similar values of the target variable.

The primary goal of each split in a regression tree is to minimize the sum of squared residuals (SSR) in the resulting child nodes. This criterion ensures that the resulting subsets are as homogeneous as possible with respect to the target variable. The process continues in iterations, with each node potentially being split further, until a predefined stopping criterion is met, such as reaching a maximum tree depth or a minimum number of observations in each leaf.

The final model can be interpreted as a piecewise constant function, where each region of the covariate space is associated with a different constant value. This approach allows for capturing the complex, non-linear relationships between the covariates and the target variable, leading to heterogeneous predictions across different regions of the covariate space (Au, 2018; Breiman et al., 1984; Holten et al., 2024).

3.1.1 Random Forest

The Random Forest (RF) algorithm significantly improves the basic regression tree by addressing its tendency to overfit. Overfitting, a common issue where a model captures not only the underlying patterns in the data but also the noise, leading to poor generalization to out-of-sample data forecasting, is mitigated by RF. This is achieved through the use of an ensemble approach known as bagging or bootstrap aggregation, which plays a crucial role in enhancing the model's performance.

In an RF, the term forest refers to the multiple regression trees grown on different bootstrap samples of the training data. Each tree is trained independently, and the predictions are averaged to produce the final forecast. This aggregation process reduces the variance of the predictions and improves the model's performance.

Additionally, the randomness in an RF is introduced by selecting a random subset of covariates and observations at each RT, which further decorrelates the individual trees and enhances the ensemble's performance. This method is particularly effective in capturing complex, non-linear interactions among covariates, making RF a powerful tool for forecasting (Au, 2018; Breiman, 2001; Holten et al., 2024; Medeiros et al., 2021).

3.1.2 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) builds upon the principles of RT and gradient boosting to achieve high predictive performance. While regression trees provide a base model for partitioning the covariate space, gradient boosting refines this approach by combining the outputs of multiple trees sequentially, each tree aiming to correct the errors of its predecessors. Gradient boosting works by adding new trees to the ensemble sequentially, where each new tree is trained to predict the residual errors of the combined ensemble of all previous trees. XGBoost, as an implementation of gradient boosting, introduces several enhancements to improve forecasting performance of:

- Regularization: Lasso and Ridge regularization is employed to avoid overfitting.
- Hessian: Unlike traditional gradient boosting, which uses first-order derivatives (gradients) for non-linear optimization, XGBoost uses both first and second-order derivatives (Hessians), which allows for more accurate approximations of the loss function, improving the model's performance.
- Learning rate hyperparameter: XGBoost applies a learning rate to shrink the contribution of each additional tree to the aggregated estimate. This technique helps in smoothing the model and prevents overfitting by reducing the impact of individual trees (Chen & Guestrin, 2016).

3.2 Least Absolute Shrinkage and Selection Operator

The Least Absolute Shrinkage and Selection Operator (LASSO) is another regularization technique for linear regression models. LASSO improves model prediction accuracy by enforcing a constraint that shrinks some coefficients to zero, thereby performing variable selection and inducing sparsity in the covariates set.

The goal of standard linear regression is to minimize the sum of squared residuals (SSR) between the observed responses and the responses predicted by the linear approximation. LASSO modifies this approach by adding a penalty proportional to the sum of the absolute values of the coefficients. The objective function for LASSO is expressed as:

$$\min_{\beta} \left(\frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right) \tag{1}$$

where: y_i denotes the observed response, β_0 is the intercept term, β_j are the regression coefficients and x_{ij} are the covariates observations. Finally λ is a tuning parameter that controls the strength of the penalty. The inclusion of the $\lambda \sum_{j=1}^{p} |\beta_j|$ term penalizes large coefficients and forces some of them to be exactly zero when λ is of sufficient size. The choice of penalty level will be further detailed in Section 5. This results in a sparse model that includes only the most significant predictors, thus enhancing interpretability of the model and improving prediction accuracy by avoiding overfitting (Tibshirani, 2018).

4 Data

A Monte Carlo (MC) simulation will be conducted in this study to evaluate and compare several methods based on Chernozhukov et al., 2018 DML and Belloni et al., 2011 post-Lasso across various data-generating processes. Key evaluation metrics in assessing models in each scenario will be bias, variance, and proportion of violation of the 95% confidence interval. The clear advantage of this process is knowing the true DGP and allowing it to test performance against true DGP 'oracle' models. Also, it allows for the sampling of data. Thus, the treatment effect estimate and bias are estimated many times, which subsequently allows the drawing of conclusions about the behaviours of the bias. Similar studies have been previously performed; however, no systematic assessment of DML against post-Lasso models has been proposed with an extensive sensitivity analysis to the data-generating process partially linear model specifications (Fuhr et al., 2024; McConnell & Lindner, 2019; Qiu et al., 2022).

4.1 Monte Carlo simulations

For each simulation, ten competing methods will be compared, further detailed in Section 5, unless otherwise specified. The data-generating process for all simulations follows a partially linear model with continuous treatment, defined by Equation 2 and Equation 3.

$$Y = \alpha_0 T + g_0(X) + \varepsilon_1 \tag{2}$$

$$T = m_0(X) + \varepsilon_2 \tag{3}$$

make and attach diagram with this notation In Equation 2 Y is a $[nx1]$ vector representing the n observations of the outcome variable (e.g. yearly income), α_0 is the true linear treatment effect, and $g_0(X)$ is a (non)linear function of p potential controls X $[nxp]$ (e.g. age, education, et cetera). T is a $[nx1]$ vector with the treatment continuous value, determined by Equation 3, where $m_0(X)$ is another (non)linear function of controls X. Finally, ε_1 and ε_2 are noise terms. In total, around 1000 simulations with different specifications are run to assess the sensitivity of model performances in terms of mean bias of treatment effect estimate minimization. The following subsections outline and justify each simulation setup.

4.1.1 Replication simulation

Simulation 0 aims at replicating Belloni et al. (2011, Section 6.2). It sets the number of observations $n = 100$ smaller than the number of control variables $p = 200$. The simulation

samples the covariates $X \sim N(0, \Sigma)$, with variance-covariance matrix $\Sigma_{ij} = 0.5^{|j-i|}$, and the noise terms $\varepsilon_{1,i}, \varepsilon_{2,i} \overset{\text{iid}}{\sim} N(0,1)$. the linear treatment effect α_0 is set to 1, and $g_0(X) = X'\beta_0$ and $m_0(X) = X'\nu_0$ are linear functions of X. Where

$$\beta_0 = [1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, 0, 0, 0, 0, 0, 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, 0, ..., 0]', \tag{4}$$

and $\nu_{0,i} = \frac{1}{i}$, for $0 < i < 11$, 0 otherwise. This is effectively a sparse set-up, where just a few regressors have a non-0 effect on the treatment and outcome variables. On the other hand, the very high number of total regressors means that many of them are noise covariates that do not affect either the outcome or the noise variables. Also, note the presence of confounded variables and variables only related to treatment or outcome. Post-Lasso methods should perform particularly well in these sparse scenarios with high covariates dimensions (Belloni et al., 2011, 2013).

4.2 Extension simulations

The purpose of running these Monte Carlo simulations is to answer the research question by carrying out an extensive Sensitivity analysis in many dimensions of the data generating process parameters.

4.2.1 simulation 1: default simulation

Starting from the general partially linear framework defined in Equations 2 and 3, Equations 5 and 6 specify the default DGP.

$$Y = \alpha_0 T + \beta_1 X_1 + \beta_2 X_2^2 + \beta_3 X_1 X_2 + \beta_4 step(X_3) + \beta_5 X_5^3 + \varepsilon_1 \tag{5}$$

$$T = \nu_1 X_1 + \nu_2 X_2^2 + \nu_3 X_1 X_3 + \nu_4 step(X_3) + \nu_5 X_4^4 + \varepsilon_2 \tag{6}$$

By default, it means that it is the starting point for the sensitivity analysis explained in the following sections. From this simulation, one parameter at the time (1 dimension) will be changed to different values to check how the model's performances in terms of mean bias of the treatment effect affect several models defined in 5 (Fuhr et al., 2024). The number of observations is set to $n = 200$. Here, one can notice that both $g_0(X)$ and $m_0(X)$ are non-linear functions of controls X, for example, the polynomial terms and the step function. The step function is defined by drawing a $u \sim U[0,1]$, if $u < \frac{1}{3}$, then $step(a) = -a$, if $u > \frac{1}{3}$, then $step(a) = a$, and finally for the remaining $\frac{1}{3}$ of the cases $step(a) = 0$. Furthermore, there is a high degree of confounding between treatment T and outcome Y via X_1, X_2, X_3. There are also two variables, X_5, X_4, that just affect the outcome and the treatment, respectively. Finally, we assume the researcher observes other variables $X_6, X_7, ...X_{20}$ that have no effect, but they suspect they may have due to the X correlation structure or the economic nature of the variables, and therefore 15 these noise controls are (mistakenly) included in X. In the sparsity framework, thus, we have a number of controls $p = 20$, and a number of non-sparse (non-0 effect) controls $s + m_y + m_d = 6$. Confounding variables number denoted by $s = 4$, and $m_y = m_d = 1$ refer to the variables X_5, X_4 just affecting outcome y and treatment d respectively. The $[20x1]$ vectors of

parameters β, and ν are defined as: $\beta_i = \frac{1}{i}$, for $i < 6$, $\beta_i = 0$ otherwise, $\nu_i = \frac{1}{i^2}$ for $i < 6$, $\nu_i = 0$ otherwise. As for the replication simulation, the covariates matrix X, of dimensions $[200x20]$ is sampled from $X \sim N(0, \Sigma)$, with variance-covariance matrix $\Sigma_{ij} = 0.5^{|j-i|}$, and the noise terms $\varepsilon_{1,i}, \varepsilon_{2,i} \overset{\text{iid}}{\sim} N(0,1)$. The true linear treatment effect α_0 is also set to 1. Finally, the number of MC simulation iterations will be 300 for this simulation and all the further ones instead of the thousand used in the replication simulation of Section 4.1.1. This choice is motivated in the following Section 4.2.2.

4.2.2 simulations 1.1-1.21: tuning number of simulations

These simulations aim to tune the number of MC simulation iterations. The reason for this is to deal with the trade-off between computation time and complete information about the treatment effect bias distribution of a given model. To solve this, 21 'default' simulations described in Section 4.2.1 are run with a number of iterations:

$$n_{sim} \in \{5, 10, 15, 20, 25, 50, 75, 100, 125, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000\} \tag{7}$$

The mean of the competing models is plotted against the number of iterations with the aim of finding a cutoff level, after which the number of iterations does not add gains in terms of stability of the bias estimates. To make this more robust, the same analysis is carried over using the median bias, which is more robust to outliers and, together with the mean, can give an indication about the third moment of the bias distribution. The outcome of this analysis will be used to set the number of iterations in all simulations ran, except the replication simulation 0 that will be kept at 1000 iterations for consistency with Belloni et al., 2011, Section 6.2 (see Section 4.1.1).

4.2.3 simulations 2.1-2.10: Sensitivity to sample size

The purpose of running these ten simulations is to verify what happens to the bias results of the default simulation 1 (Section 4.2.1) when the sample size changes. For this aim, median bias, standard deviation and proportion of 95% confidence interval violations are computed for the following values of observations:

$$n \in \{50, 100, 200, 300, 400, 500, 1000, 2000, 4000, 10000\} \tag{8}$$

We expect all models to perform better with more observations. Regarding the relative performance of the models, the DML tree-based methods DML-RF and DML-XGBoost should particularly gain from higher observations, as they can better 'learn' the non-linear specifications from a linear set of covariates (see Section 5). While Post-lasso methods may perform relatively worse as they are intended for high-dimensional settings (Belloni et al., 2013, 2016).

4.2.4 simulations 3.1-3.10: Sensitivity to noise covariates

These ten simulations tune default simulation 1 with different amounts of noise variables (number of noise variables $= p - s$). The values chosen for noise variables examined are:

$$(p - s) \in \{0, 5, 10, 15, 30, 50, 100, 200, 500, 1000\} \tag{9}$$

We expect all models to perform worse in absolute terms with more noise variables. In terms of relative performance of the models, LASSO-based methods from the Belloni et al., 2011 replication such as double selection, and DML-LASSO should perform particularly well with a higher amount of noise variables as they are designed for approximate sparsity in high dimensional datasets. Methods based on multiple LS will not be able to run for $(p - s) \in 200, 500, 1000$, as this would violate the rank condition for the X regressors.

4.2.5 simulations 4.1-4.10: Sensitivity to confounding strength

Let us rewrite Equation 5 and 6 of the default simulation 1, including an extra term $\phi > 0$ of confounding strength. Notice that under the case $\phi = 1$ Equation 10 and 11 are equivalent to the default simulation 1 scenario

$$Y = \alpha_0 T + \phi(\beta_1 X_1 + \beta_2 X_2^2 + \beta_3 X_1 X_2 + \beta_4 step(X_3)) + \beta_5 X_5^3 + \varepsilon_1 \tag{10}$$

$$T = \phi(\nu_1 X_1 + \nu_2 X_2^2 + \nu_3 X_1 X_3 + \nu_4 step(X_3)) + \nu_5 X_4^4 + \varepsilon_2 \tag{11}$$

Thus simulation 1 of Section 4.2.1 bias results are analyzed under 10 different confounding strength ϕ values:

$$\phi \in \{0.1, 0.5, 1, 2, 4, 6, 8, 10, 15, 20\} \tag{12}$$

All models are expected to perform worse in terms of absolute mean bias minimization under stronger confounding ϕ. In particular, those who do not take into account confounding in the first place, such as the benchmarks simple-OLS, naive-OLS, and the Feasible LASSO method, should result in particularly poor performances.

4.2.6 simulations 5.1-5.10: sensitivity to numbers of confounders s

This simulation adapts the default simulation 1 to relax the value of confounders $s = 4$ and consider ten different confounders values. To keep the number of noise variables $p - s - m_y - m_d = 14$ constant p X number of covariates also changes accordingly to number of confounders s in simulations. The confounders values simulated are:

$$s \in \{10, 20, 30, 40, 50, 100, 150, 200, 250, 300\} \tag{13}$$

Notice again that this means that some methods will be unfeasible for higher parameters due to X regressors $p > n$ number of observations (rank condition of OLS violated).

For every simulation scenario and iteration, we need to add other X columns as regressors for both outcome Y and treatment T. In Equations Equation 5 and 6 of the default simulation, a variable can enter the nonlinear $g_0(X), m_0(X)$ in many ways. To not preclude any possibility, a

random functional form will be decided by allocating equal probability to the following functional forms for each additional regressor: *i.* linear: $X_{i,s} = X_i$, *ii.* squared: $X_{i,s} = X_i^2$, *iii.*: $X_{i,s} = X_i^3$, *iv.*: $X_{i,s} = X_i^4$, *v.*: $X_{i,s} = X_1' X_i$, *vi.*: step function, as described in Section 4.2.1.

In the sensitivity to number of confounders, all methods are expected to perform worse the more confounders we add. However it is unclear which perform relatively best under this scenario. Simulations 5.1 to 5.10 help clarify this.

4.2.7 simulations 6.1-6.10, and 7.1-7.10: Sensitivity to number of variables that only affect the outcome and the treatment

Recall how in the default simulation 1 detailed in Section 4.2.1, variable X_4 just explains outcome Y, thus $m_y = 1$. Also, $m_d = 1$, as just X_5, is related to the outcome T but is not confound with Y. In simulations 6.1 to 6.10, we expand the default simulation to test the sensitivity of bias of the models to m_y, the number of variables just related to the outcome. In simulations 7.1 to 7.10, an analogous process is carried over, this time holding m_y fixed to 1 and trying different values of the number of variables just related to treatment m_d. The values attempted are:

$$m_y, m_d \in \{0, 2, 4, 6, 8, 10, 20, 30, 40, 50\} \tag{14}$$

Except for the cases $m_y = 0, m_d = 1$, or vice versa $m_y = 1, m_d = 0$, we need again to add other X columns as regressors of either outcome or treatment. To do this, the random functional form described in Section 4.2.6 is employed, with a small adjustment to the step function described in Section 4.2.1 where the probability $\frac{1}{3}$ of the cases $step(a) = 0$ is removed to make sure the variables are actually selected for explaining Y or T. Thus, the step function in this case is defined as: draw a $u \sim U[0, 1]$, if $u < \frac{1}{2}$, then $step_{\neq 0}(a) = -a$, otherwise $step_{\neq 0}(a) = a$.

The expectation is for simulations 7.1 to 7.10 to reveal great changes in relative performances as some model do not fully take into account the T process (such as the methods post-LASSO, feasible LASSO, naive-OLS, simple-OLS described in Section 5). Also note that these variables could be considered as IV (see Section 3.

4.2.8 simulations 8.1-8.10: sensitivity to unobserved confounders

Up until now we assumed that all relevant covariates are observed. In this simulation we relax the assumption of orthogonality $\mathbb{E}(X'\varepsilon) = 0$ by purposefully omitting confounder variables. The number of unobserved confounding variables both related to X and Y are denoted by u_x, with:

$$u_x \in \{1, 2, 4, 6, 8, 10, 20, 30, 40, 50\} \tag{15}$$

The functional form of the new covariates is determined by the random (nonlinear) function described in the section right above. Note that the functional form that a certain confounder X_i takes in relationship to the outcome Y is chosen independently than the functional form X_i takes in relation to treatment T.

In the sensitivity to number of unobserved confounders, all methods are expected to perform worse the more confounders are omitted. However it is unclear which perform relative best under

this scenario. The need to explore the relative performance of model is also motivated by the likelihood in practical applications of omitting confounders even in high dimensional datasets.

4.2.9 simulations 9: including interaction and squared terms

In Section 2, in the context of IV, we outlined how Belloni et al. (2014) in high-dimensional datasets with more controls than observations finds unintuitive instruments consisting of higher-order interaction terms to be stronger than instruments derived solely by considering linear IVs and linear controls. In these simulations we extend this concept to the replication simulation and default simulation 1.

Lets take simulation 1 for example. Note that the DGP for $m_0(X), g_0(X)$ is non linear. Squared and interaction terms appear in the specifications. Therefore the intended purpose is to allow the models to better estimate more exact non-linear models to estimate the bias more accurately (reduce absolute mean bias). In simulation 1 we have $n = 200$ observations and $p = 20$ covariates. We augment the model by considering a model with squared terms in addition to linear terms, so $p = 2 \times 20 = 40 < n$, and a model with interaction terms in addition, so $p = 210 = 20 + (19 + 18 + 17 + ... + 1) > n$. Finally we consider a case in which both squared and interaction terms are considered, with linear terms: $p = 20 + 20 + 190 = 230 > n$.

The same simulations are ran for the replication simulation. However here there is a complication. If we were to augment the replication simulation with the addition of interaction terms we would have: $p = 200+199+188+...+1 = 20'100 >> n = 100$, This is not a problem for most of the methods. However such a high number of dimensions makes the methods very inefficient as a lot of sparse noise variables are included, and as computation times increases considerably. Therefore for this simulation a new replication simulation is considered with all the same as the one in Section **??** except for the number of covariates $p = 20$. Therefore there are 180 less noise covariates, leading to a great reduction of interaction terms additional covariate dimensions to consider. The same olds for the replication simulation with squares and interaction terms in addition to linear combinations. While for the methods with linear terms and squares, the p dimension just doubles, and therefore the original replication simulation setup with $p = 200$ is maintained in this instance.

4.2.10 simulations 10: hyper-parameter tuning of RF

Finally we run another additional method that is DML-RF-Tuned on repliacation simulation and default simulation 1. In this simulation we tune the 2 RF algorithm at each Monte Carlo iteration to better forecast the residuals. The hope is better performances, due to lower regularization bias in the DML procedure described in Section 3. The aim of these simulations is to compare DML-RF-Tuned to DML-RF to verify if it is indeed bias performance improving. Note that performing hyperparameter tuning in a monte carlo setting results extremely computationally heavy. For this reason this extension method is not performed on other DML algorithms not on sensitivity simulations. The procedure used consists in tuning jointly the RF hyperparameters: number of covariates to consider in each tree, minimum number of observations in each node (stopping condition), random fraction of sample to consider in each tree. Other hyperparameters such as the number of trees=200 are not tuned but kept to default values. The tuning checks 200

combinations of RF hyperparameters, 100 for each RF forecasting task. The first 30 iterations perform a random serch around the grid of value to begin to approach the non-linear optimization problem of fiding the hyperparameter that minimize the root mean squared error (RMSE), a measure of forecast errors. The remaining 70 iterations are performed using the `mlrMBO` which implements the Efficient Global Optimization Algorithm based on Bayesian optimization(Bischl et al., 2017).

5 Methodology

5.1 Post-Lasso estimation

In Section 3, the LASSO operator was defined. Here, we expand on its implementation in this research by considering different possible specifications for the penalty term λ:

Default method LASSO penalty *Feasible LASSO Penalty* The feasible LASSO method employs a penalty term that is data-driven, optimizing the balance between bias and variance to minimize prediction error.

X-dependent LASSO Penalty Belloni et al. (2011, 2013) introduced an X-dependent penalty that adjusts for the dimensionality and correlation structure of the covariates. This method ensures that the penalty term adapts to the complexity of the data, providing more robust variable selection.

Default Method LASSO Penalty The default method LASSO penalty, implemented as a standard in most LASSO regression packages, typically uses cross-validation to determine the optimal λ. This approach ensures a general and automated selection process suitable for various datasets.

Post-LASSO Procedures *i. Feasible LASSO Method*: Applies the feasible LASSO penalty to select relevant variables, which are then used in subsequent regression models.

ii. Post-LASSO: After selecting variables with LASSO, a standard OLS regression is performed using the selected variables to obtain unbiased coefficient estimates.

iii. Indirect Post-LASSO: Similar to Post-LASSO, but the selection of variables is indirectly refined through additional criteria before OLS regression.

iv. Double Selection: Combines LASSO for both the outcome and treatment equations to ensure robustness against model selection errors.

v. Double Selection Oracle: An idealized version of double selection, assuming perfect knowledge of the true model, providing a benchmark for evaluating other methods.

vi. OLS Oracle: A benchmark OLS model assuming perfect knowledge of the relevant variables, used for comparison against other methods.

5.2 Double Machine Learning Methods

Double Machine Learning (DML) models are based on Chernozhukov et al., 2018 and Fuhr et al., 2024. These models incorporate hybrid approaches combining Belloni et al. (2011) methods with DML techniques. Notably, the double selection method can be seen as a specific DML-Lasso variant with X-dependent λ penalties.

vii. DML-Lasso: Uses LASSO for both outcome and treatment models, followed by cross-fitting to control for overfitting and ensure unbiased treatment effect estimates.

viii. DML-Random Forest (DML-RF): Constructs multiple decision trees using random subsets of covariates, averaging their predictions to reduce variance and improve robustness.

ix. DML-XGBoost: Employs the XGBoost algorithm for gradient boosting, enhancing prediction accuracy through regularization and second-order derivative optimizations.

x. DML-OLS-Oracle: An OLS model within the DML framework, assuming perfect variable selection to serve as a performance benchmark.

xi. DML-Lasso-Oracle: A DML model with idealized LASSO variable selection, used as a benchmark for evaluating practical LASSO implementations.

xii. DML-Lasso-Belloni-penalties: Utilizes the X-dependent λ penalties specified by Belloni et al., 2013 within the DML framework for more precise variable selection and estimation.

5.3 Other Extension Benchmarks

xiii. Simple OLS: A basic OLS regression of outcome on treatment, expected to produce biased estimates due to omitted variable bias.

xiv. Naive OLS: An OLS regression including all covariates, expected to be less biased than simple OLS but potentially inefficient due to noise variables. In high-dimensional scenarios where p¿n, this method becomes infeasible due to the non-invertibility of X'X.

Models are evaluated by comparing treatment effect bias characteristics: mean bias, bias standard deviation and rejection proportion of the 95% confidence interval. The SE used for computing the (normal) confidence interval are White's Heteroskedasticity-Consistent Standard Errors implemented via the jackknife procedure (MacKinnon & White, 1985).

The analysis utilized the following software and packages:

- `hdm`. Used for Post LASSO variable selection, residual calculations, and penalty estimations for DML-Lasso-Belloni-penalties, as well as for double selection and indirect Post LASSO methods (Chernozhukov et al., 2016).
- `DoubleML, ranger, xgboost`. Used for implementing DML models (Bach et al., 2024).

6 Results

6.1 Replication results

Figure 20 in the appendix reports the results from Belloni et al. (2011, Section 6.2) which this study aims to replicate and compare to other DML and benchmark methods introduced in Section 3. Table 1 contains the results for the replication methods and the extension methods and benchmarks. Looking at the models ranked in terms of absolute mean bias in Figure 20 and Table 1,the results seem to align with the original paper in terms of order of magnitude of the estimated mean biases. However there are some differences in the ranking of the estimation methods based on their bias. While feasible LASSO results in the worst performing model and Oracle-OLS in the best in both studies and also the results for double selection oracle, post-LASSO results in relative lower biases in this replication study. Also note that the double

Table 1: Performance of 13 models over 1000 monte carlo iterations in Belloni et al. (2011, Section 6.2) simulation setup

Model	**Mean Bias**	**Std Dev**	**rp(0.05)**
naive-OLS	NA	NA	NA
simple-OLS	0.3487	0.1572	0.9300
DML-Lasso-Belloni-penalties	0.3487	0.1573	0.7880
DML-RF	0.2990	0.1439	0.7470
Double selection	0.2990	0.1338	0.5210
DML-XGBoost	0.2981	0.1370	0.7610
Lasso	-0.2873	0.2751	0.5490
Indirect Post-Lasso	0.1976	0.1423	0.4960
DML-Lasso	0.1872	0.1280	0.5110
Double selection Oracle	0.1872	0.1321	0.4900
Post-Lasso	-0.1289	0.2416	0.3560
DML-oracle-OLS	-0.0271	0.1322	0.0120
Oracle-OLS	0.0012	0.0589	0.0590

Note. naive-OLS is unfeasible (number of covariates bigger than number of observations).

selection seems to get a bias that is too high and an incorrect rejection rate for the 95% coverage in this thesis replication, this holds more in general for the majority of models. The potential reasons for the discrepancy should be further investigated as it may be occurring due to a mistake in the implementation in this paper of the jackknife standard errors to compute the rejection rates, in the models implementation, or a lack of replicability in Belloni et al., 2011. Despite no code or data being provided with the paper, the results are feasible to replicate as the methods and simulation setup are described at length (except the oracles), and there is `hdm`, a `R`package based on these methods that aids in the replication (Belloni et al., 2011, 2013; Chernozhukov et al., 2016).

Regarding the extension models introduced for comparison in Table 1, it can be noted that in this particular replication of the Monte Carlo DML-Lasso simulation is the best performing model, achieving near oracle double selection performances. The worst performing DML model is the hybrid DML-Lasso-Belloni-penalties, not faring much better than the simple-OLS benchmark which is heavily biased due to omitted variable bias. The DML-RF also underperforms in terms of mean bias compared to the Belloni et al. (2011, Section 6.2) replication methods. However this is to be expected due to the linear specification of this simulation detailed in Section 4 and the RF being good at forecasting in a highly non-linear method. The underperformance could also be due to a lack of hyperparameter tuning, and this will be further investigated in Section 6.2.10.

6.2 Extension simulations results

6.2.1 simulation 1.1-1.21: tuning number of Monte-Carlo iterations

The objective of this section is to explore the impact of varying the number of Monte Carlo (MC) iterations on the mean bias of the estimation methods employed in this study. The results of this analysis, focusing on mean bias, are detailed in Figures 3 and 4.

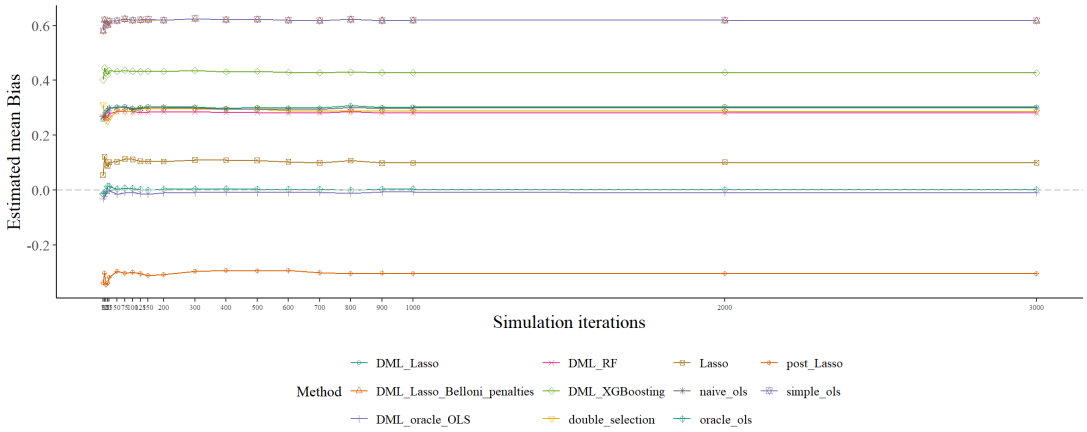

Figure 3: Mean bias for 11 competing models (in legend) against number of MC simulation iterations

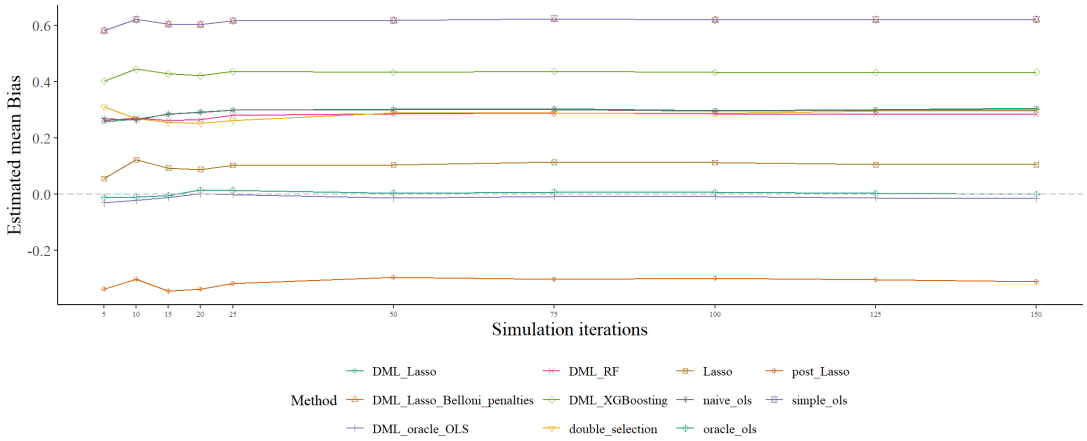

Figure 4: Mean bias for 11 competing models (in legend) against number of MC simulation iterations. Selection of iterations up to 150.

Upon examination of the figures, it becomes evident that increasing the number of MC iterations beyond a certain threshold does not yield significant additional improvements in bias reduction. Specifically, the bias tends to stabilize after approximately 100-150 iterations. This finding suggests that while initial iterations are crucial for achieving reliable estimates, excessively high numbers of iterations may lead to diminishing returns in terms of bias reduction.

To strike a balance between accuracy and computational efficiency, this study adopts a practical approach of conducting 300 iterations for each Monte Carlo simulation, excluding the replication process. This decision ensures robustness in the estimation results while optimizing computational resources. For a comprehensive assessment, Figures 21 and 22in the appendix, present similar insights focusing on the median bias across varying numbers of MC iterations. These supplementary results reaffirm the observed stabilization in bias reduction beyond a certain threshold, validating the chosen value of 300 iterations as a suitable compromise.

The analysis conducted concluded that the preferable number of MC iterations is 300. Therefore this value is used to estimate the bias metrics reported in Table 2. The models are ordered

Table 2: Default simulation 1 bias metrics results.

Model	Mean Bias	Standard Deviation	rp(0.05)
oracle OLS	0.0057	0.0709	0.0633
DML-oracle OLS	-0.0062	0.0778	0.0600
DML-RF	0.0871	0.0862	0.1633
naive OLS	0.0901	0.0901	0.2100
DML-LASSO	0.0914	0.0903	0.1867
DML-XGBoosting	0.1132	0.0852	0.2000
simple-OLS	0.2518	0.0907	0.8533
DML-LASSO-Belloni-penalties	0.2518	0.0907	0.7733
double selection	0.2943	0.1106	0.7867
post LASSO	-0.5610	0.1079	0.9700
LASSO	-0.5888	0.1319	0.9800

by lowest to highest in terms of absolute mean bias of treatment effect. In this default simulation setup the model that produce the most bias are feasible LASSO, followed by post-LASSO and double selection. The best performing models are DML-RF, naive OLS, and DML-LASSO. These results are important as this simulation setup serves as the reference point on which all future sensitivity simulations are built upon.

6.2.2 simulations 2.1-2.10: Sensitivity to sample size

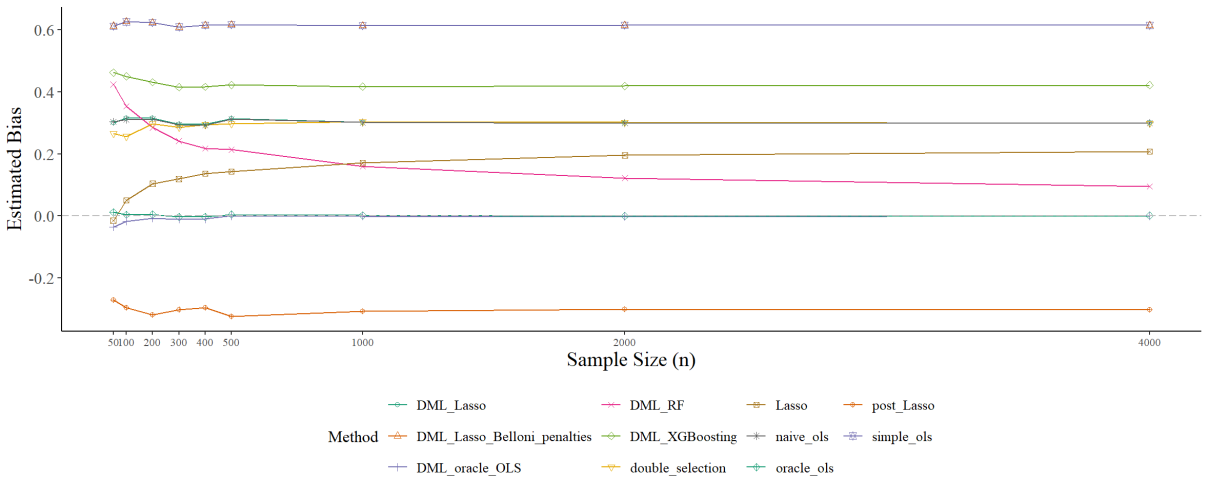

Figure 5: Mean bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of observations

In this section the results concerning the sensitivity analysis of the default simulation to changes in sample size are reported. Firstly, concerning mean bias, as depicted in Figure 5, contrary to initial expectations, the majority of methods do not exhibit consistent improvement in mean bias with increasing sample size. Instead, the mean bias tends to stabilize around 1000 observations for most models.

Notably, DML-RF and feasible LASSO stand as an exception. Unlike most competing mod-

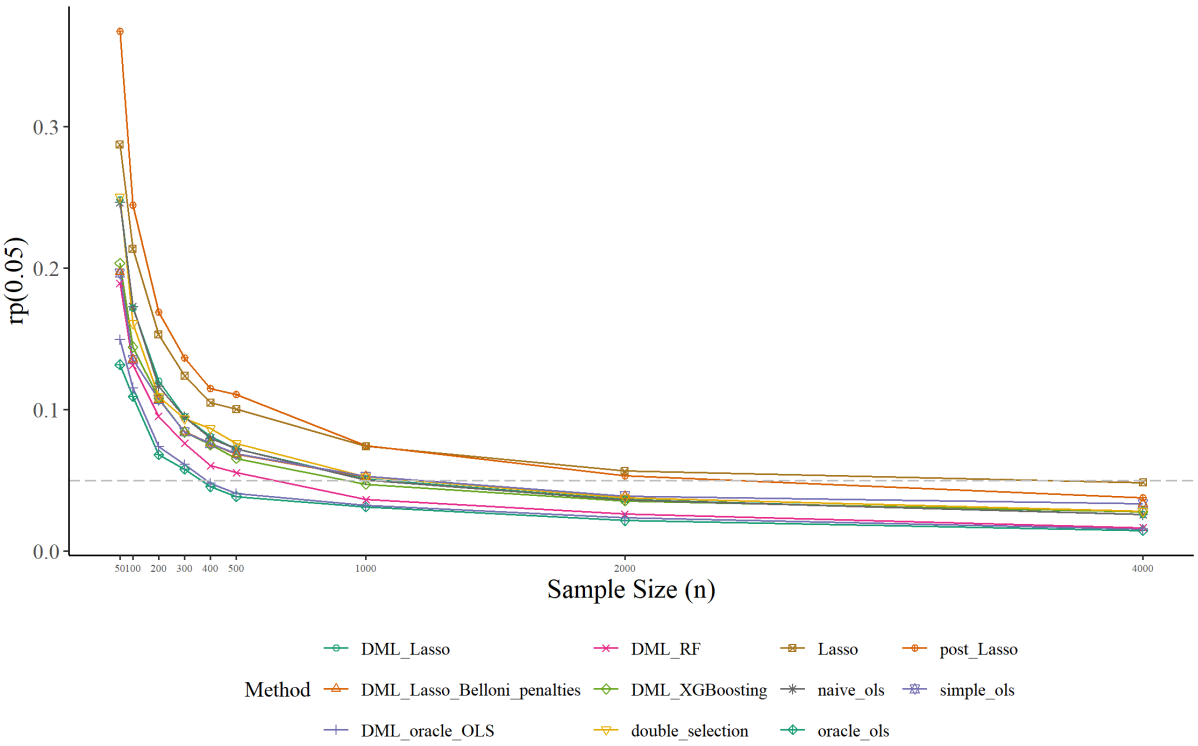

Figure 6: 95% rejection rate for the estimated bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of observations

els, DML-RF manifests a progressive improvement in performance as the sample size grows. This increase is attributed to its robust capacity to capture the non-linear specifications inherent in $g_0(X)$ and $m_0(X)$. Feasible LASSO also deserves particular attention in the analysis. At a sample size of $n = 50$, it demonstrates near oracle performance, indicative of highly accurate treatment effect estimations. However, with increasing sample size, a concomitant rise in absolute mean bias for Feasible LASSO is observed. This contrasts sharply with the sustained performance improvement of DML-RF with larger sample sizes. The graphical representation in Figure 5 portrays these dynamics, evidencing that Feasible LASSO outperforms DML-RF for $n < 1000$, while DML-RF becomes increasingly preferable beyond this threshold.

Turning to bias standard deviation and rejection frequency, as depicted in Figure 23 in the appendix, and in Figure 6, both metrics exhibit analogous patterns across varying sample sizes. Notably, an increase in sample size n corresponds with a conspicuous decrease in both bias standard deviation and rejection frequency. This observed phenomenon is explicated by the proportional relationship involving n in the denominators of standard deviation and standard error definitions, thereby influencing rejection frequency accordingly. Larger sample sizes contribute to enhanced precision in estimates and diminished variability across iterations. It should also be noted that the sample size could be tuned appropriately to make the rejection frequency of the confidence interval correct for the desired model. Therefore it is recommended that future research includes sensitivity to sample size for robustness instead of just reporting one arbitrary value.

6.2.3 simulations 3.1-3.10: Sensitivity to noise covariates

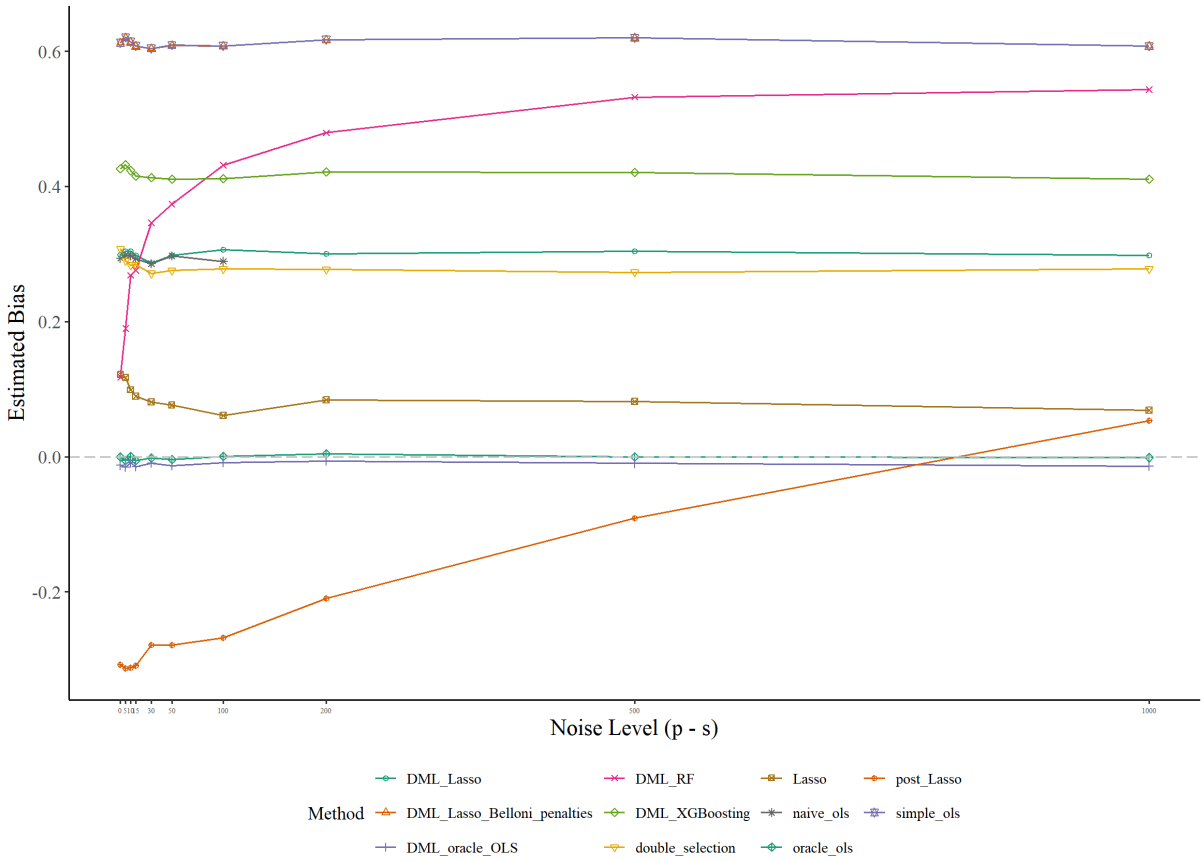

Figure 7: Mean bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of noise covariates

Figure 7 presents the mean bias results for 11 competing models under varying numbers of noise covariates. Contrary to expectations, the majority of methods exhibit very low sensitivity to an increase in the number of noise covariates.

Noteworthy exceptions include the post-LASSO and DML-RF methods, both of which demonstrate a pronounced increase in treatment effect estimates as the number of noise covariates proliferates. Specifically, DML-RF exhibits a logarithmic increase in treatment effect estimates with noise covariates number exceeding 100. Beyond this threshold, it performs less favorably in terms of mean bias compared to all other competing models, except DML-LASSO-Belloni-penalties and simple OLS, which are heavily positively biased across the entire domain of noise variables considered.

The standard deviation of bias and rejection frequency, as displayed in Figure 24 and Figure 25 in the appendix, do not exhibit discernible patterns in response to the inclusion of noise covariates in the X covariates. It is pertinent to note again that the naive OLS benchmark is rendered inapplicable when the number of noise variables reaches or exceeds zero, due to violations of the rank condition.

In summary, the empirical findings underscore varying degrees of robustness among estimation methods to increasing numbers of noise covariates. While most methods remain resilient

to such perturbations, Post Lasso and DML-RF demonstrate notable sensitivity, necessitating careful consideration in empirical high-dimensional applications.

6.2.4 simulations 4.1-4.10: Sensitivity to confounding strength

Figure 8 presents the mean bias for 10 different models under varying degrees of confounding strength, denoted by the constant ϕ as defined in Section 4. Note that post-LASSO as been omitted as it behaved very poorly and ruined the scale for interpreting the other models. The complete results are found in the appendix in Figure 26The analysis reveals that all methods, except the oracle benchmarks, experience deteriorating performance as confounding strength increases. Initially, the performance of all methods declines sharply with increased confounding, but beyond a certain threshold, the rate of decline becomes marginal. Different methods reach this threshold at varying points, leading to distinct results regarding performances in strong confounding scenarios.

Feasible LASSO consistently minimizes treatment effect estimation across the entire sample. For $phi \geq 1$ it shows the lowest mean bias among it is the best performing model. For low confounding levels (up to $\phi = 2$) performance across models is relatively similar. However, as confounding strength increases, model performance diverges. Double selection and naive OLS form a low-performing set under strong confounding conditions. In contrast, all other non-oracle models, particularly the DML methods, perform better and are recommended for practitioners in such settings. Simple OLS outperforms DML methods, with Feasible LASSO outperforming all.

Figure 9 and 27 in the appendix report the bias standard deviation for the competing models. The standard deviation generally shows little sensitivity to confounding strength, except for the oracle benchmarks DML-oracle OLS and oracle OLS, where uncertainty around bias and treatment effect estimates increases with higher confounding strength. These oracle models also perform best in terms of rejection rate, as shown in Figure 28 in the appendix. Other models reject too often, indicating that model misspecification (omitting non-linear regressors in X may be affecting standard errors and rejection rates.

6.2.5 simulations 5.1-5.10: sensitivity to numbers of confounders s

This set of simulations further explores the sensitivity to confounding strength, differing from the previous assessment by increasing the dimensionality of the confounding covariates. Figures 29 and 30 (in appendix) present the mean bias and bias standard deviation for 11 models under varying numbers of confounders. The oracle-OLS model exhibits unusual behavior, with mean bias and bias standard deviation increasing sharply once p¿n. Consequently, we exclude oracle-OLS from comparative analysis, focusing on the remaining 10 models in Figures 10, and 11. Figure 31 displays the 95% confidence intervals for all 11 models.

Notably, naive OLS is infeasible for $p > n$ due to the violation of the rank condition, rendering $X'X$ non-invertible. Naive OLS underestimates the treatment effect more severely as the number of confounders increases. DML-RF performance also degrades with additional confounders, likely because the random variable selections become less informative about the true data-generating process. Contrary to Belloni et al., 2011, Feasible LASSO outperforms Post-LASSO, though

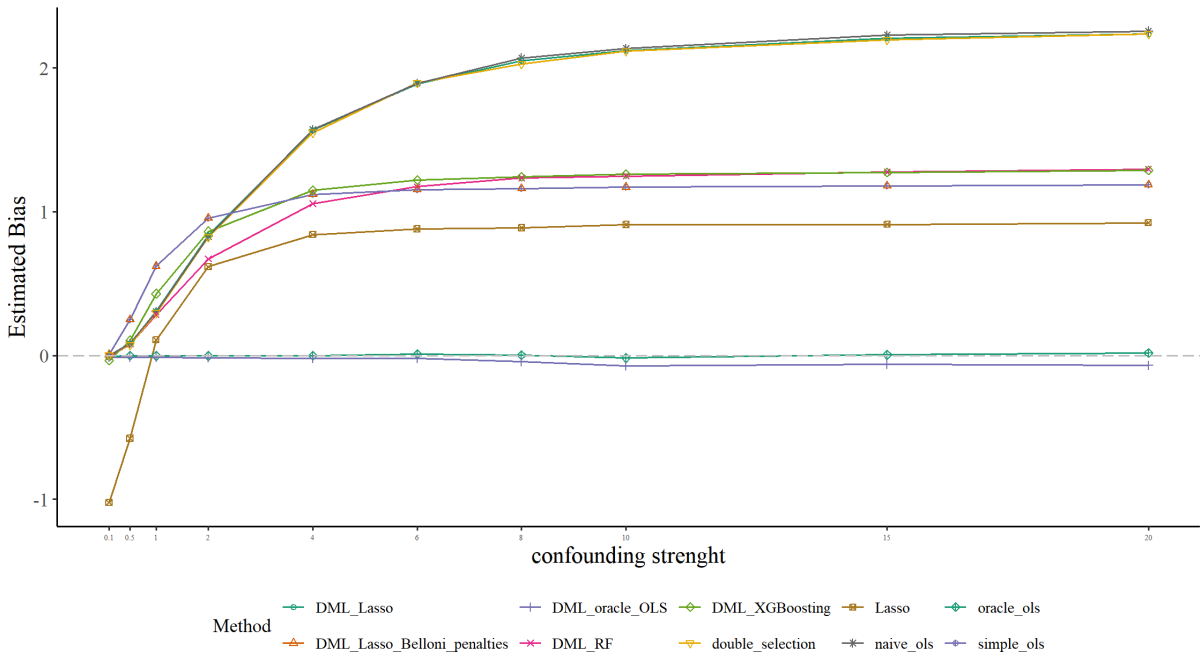

Figure 8: Mean bias for 10 competing models (in legend), over 300 Monte Carlo iterations, for different confounding strength multipliers

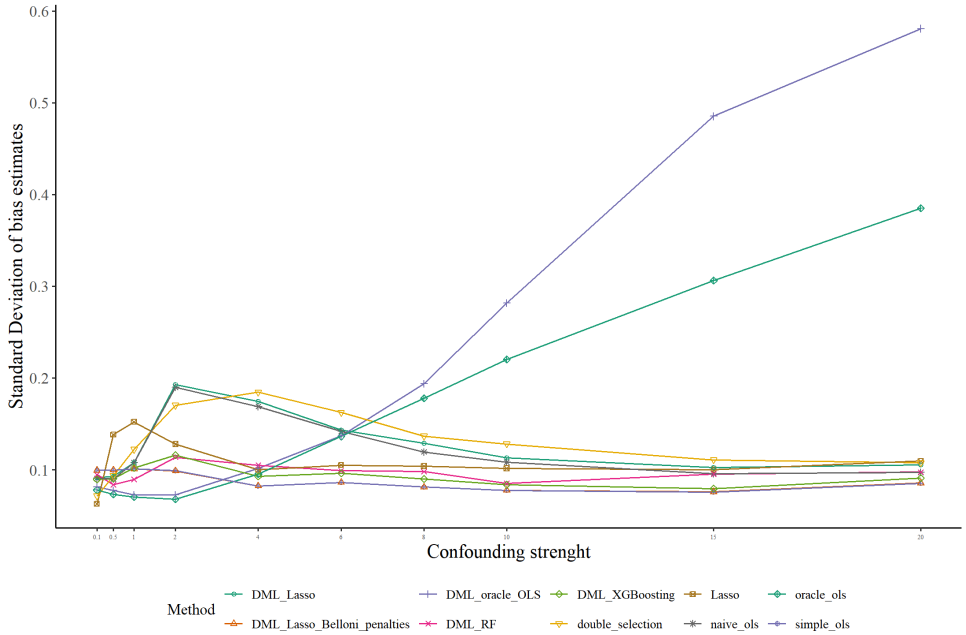

Figure 9: Standard deviation of the bias for 10 competing models (in legend), over 300 Monte Carlo iterations, for different confounding strength multipliers

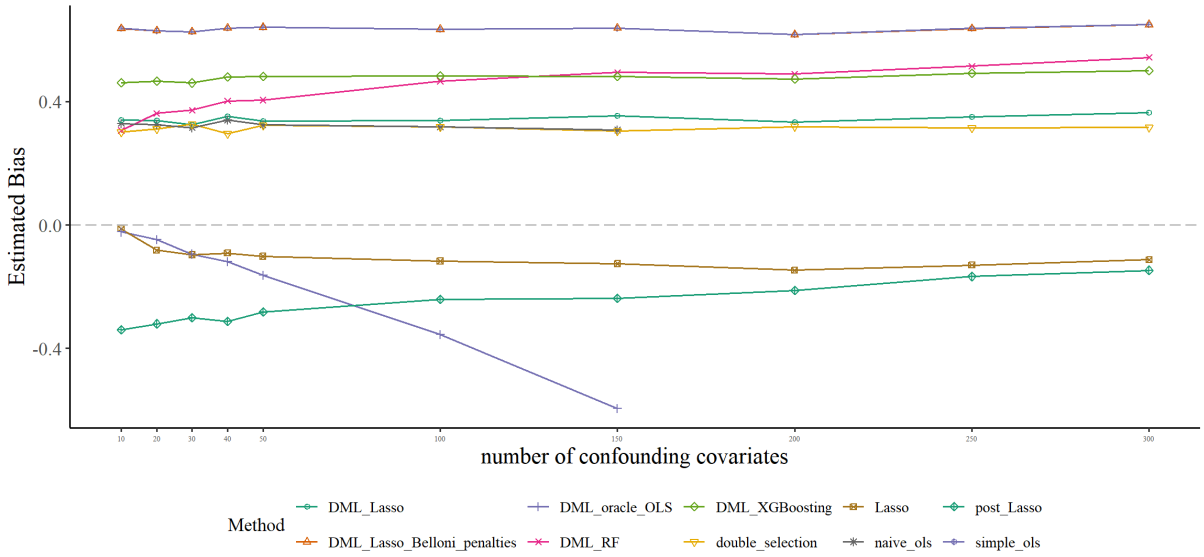

Figure 10: Mean bias for 10 competing models (in legend), over 300 Monte Carlo iterations, for different number of confounding covariates

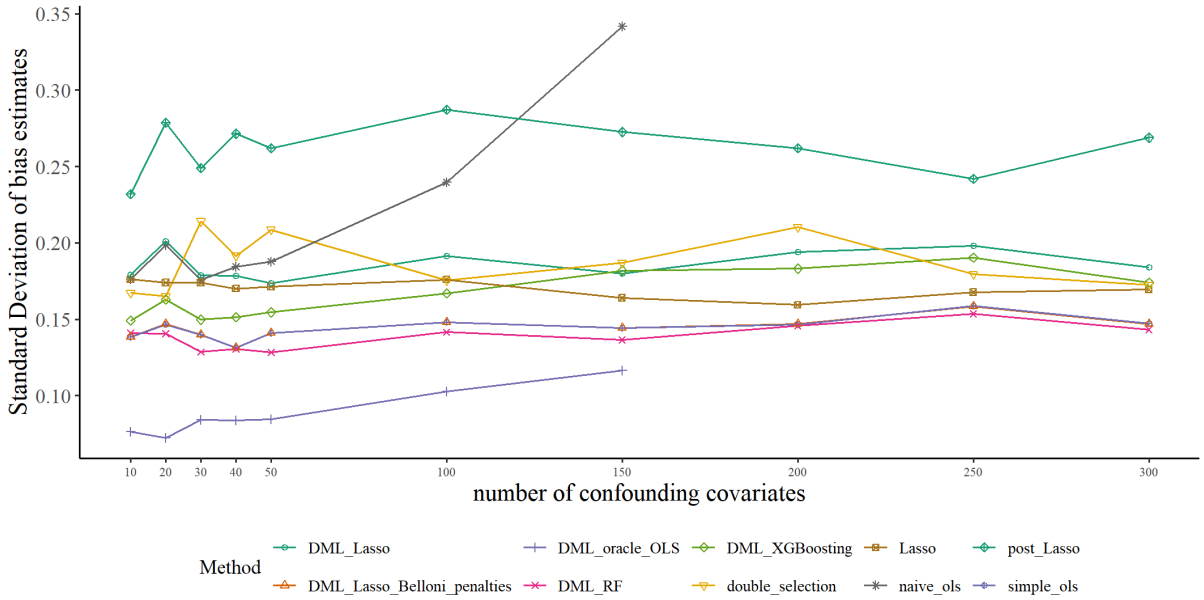

Figure 11: Standard deviation of the bias for 10 competing models (in legend), over 300 Monte Carlo iterations, for different number of confounding covariates

their mean biases converge with a higher number of confounders. The other models show minimal sensitivity to the number of confounding variables.

Regarding the standard deviation of bias, OLS-based methods exhibit the expected increase in standard error as the number of controlled variables rises, explaining the higher standard deviation with more variables. Other models remain largely unaffected. Finally, in terms of rejection ratio, LASSO-based methods (LASSO, Post LASSO, and DML-LASSO) approach the 5% rejection rate, while other methods deviate significantly.

6.2.6 simulations 6.1-6.10: sensitivity to covariates that just affect the outcome

Figures 32 and 34 present the mean bias for the treatment effect estimate and the bias standard deviation for 11 models across varying numbers of covariates which affect only the outcome. DML-LASSO consistently exhibits the highest bias and the largest standard deviation. Conversely, DML-oracle OLS consistently shows the lowest bias and smallest standard deviation. Due to the substantial differences in magnitude, we focus on mean bias, standard deviation, and 95% coverage rejection rate for the remaining 9 models in Figures 12, 13, and 33.

The ranking of the best models in terms of absolute mean bias minimization changes significantly when variables affecting only the outcome are included. For the 9 models examined, mean bias performance stabilizes around eight such covariates and does not change significantly thereafter. The worst performers in this setting are LASSO and double selection, which perform worse than the simple OLS benchmark. The best performers, approaching near-oracle performance, are DML-RF and naive OLS. The superior performance of naive OLS is intuitive, as including more covariates related only to the outcome reduces bias from confounding. The reasons behind the very good performance of DML-RF relative to other inference methods, however, is less clear.

The standard deviation of bias remains relatively stable across models, with double selection being an exception due to its higher uncertainty around the bias estimate. Unlike previous simulations, here we observe that most models, except double selection and DML-oracle OLS, perform close to the prescribed 5% rejection rate.

6.2.7 simulations 7.1-7.10: sensitivity to covariates that just affect the treatment

These simulations are designed to analyze the sensitivity of our models to covariates that exclusively influence the treatment. Such covariates can be interpreted as instrumental variables (IVs), as discussed in Section 3.

Figure 14 presents the mean bias for the 11 models when varying the number of covariates that solely affect the treatment. Similar to the findings in simulations 6.1-6.10, which focused on covariates affecting only the outcome, we observe that LASSO-based models, including DML-LASSO, simple OLS, DML-LASSO-Belloni-penalties, and LASSO, exhibit the highest absolute mean bias. Interestingly, the ranking of the models in terms of minimizing absolute mean bias changes substantially with the inclusion of treatment-only covariates. However, it is noteworthy that the models appear insensitive to the quantity of such covariates; the mere presence of these covariates, rather than their number, seems to influence performance.

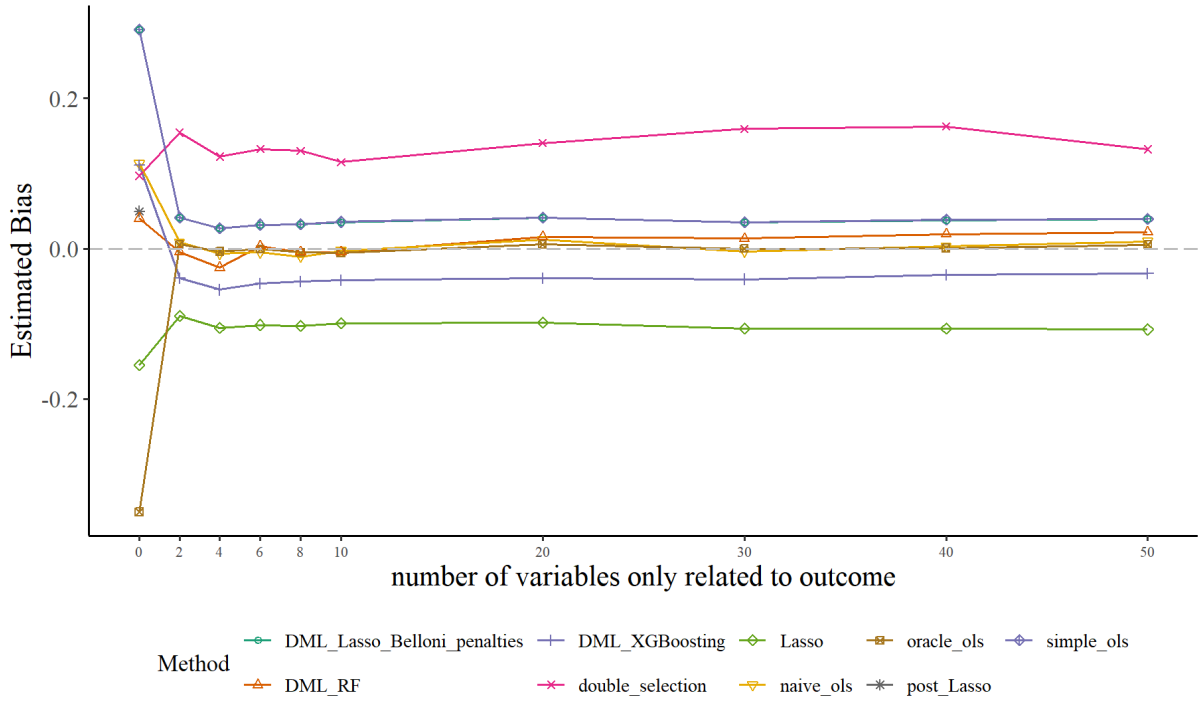

Figure 12: Mean bias for 9 competing models (in legend), over 300 Monte Carlo iterations, for different number of covariates only related to outcome

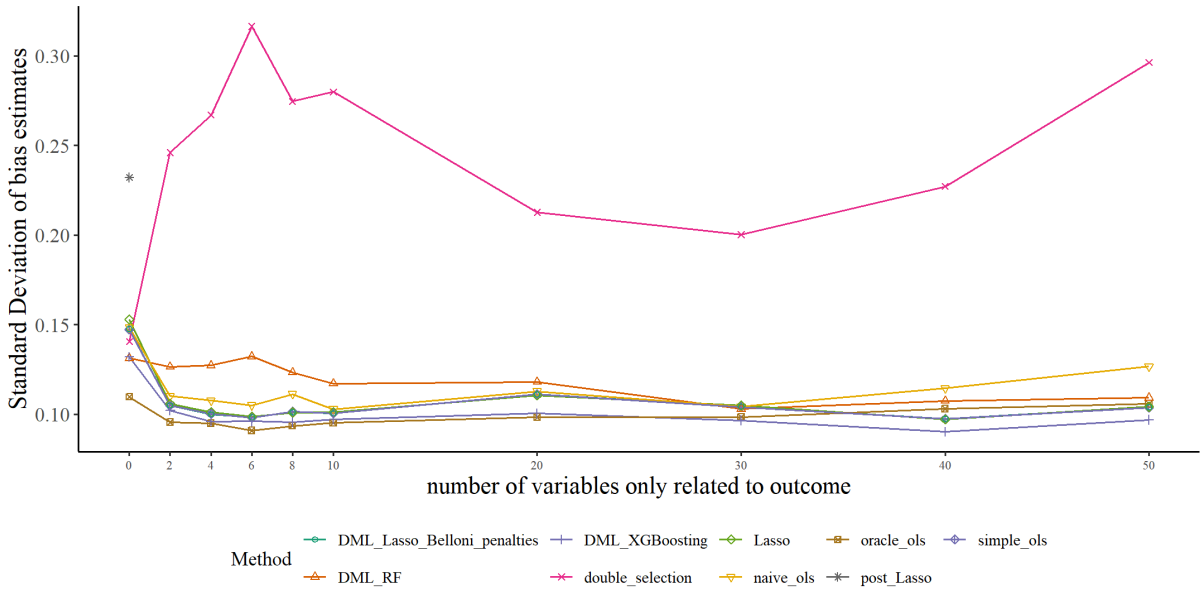

Figure 13: Standard deviation of the bias for 9 competing models (in legend), over 300 Monte Carlo iterations, for different number of covariates only related to outcome

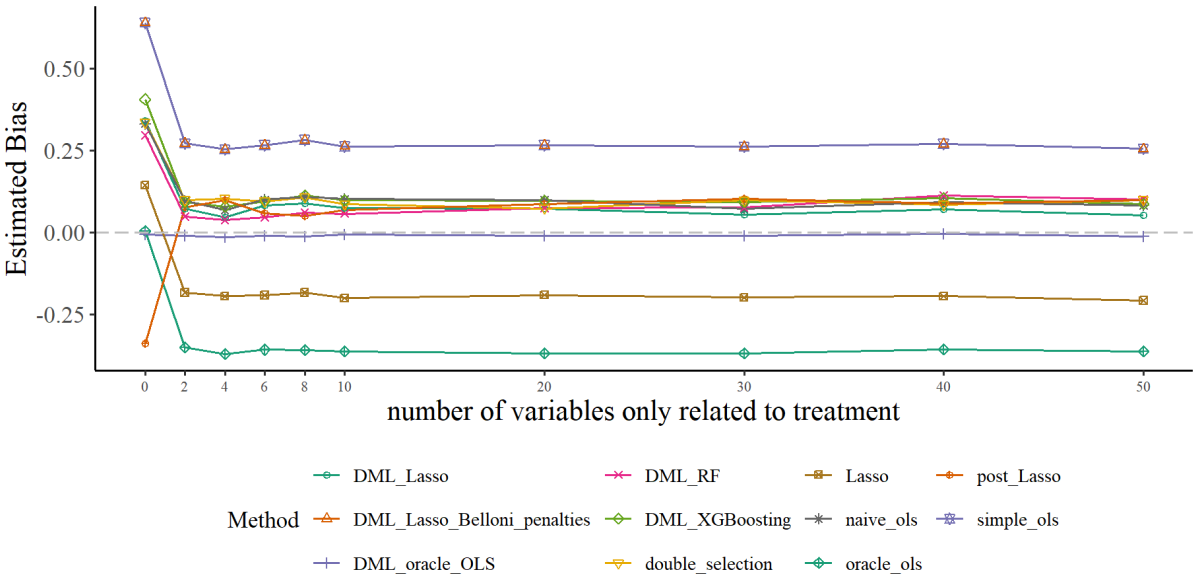

Figure 14: Mean bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of covariates only related to treatment

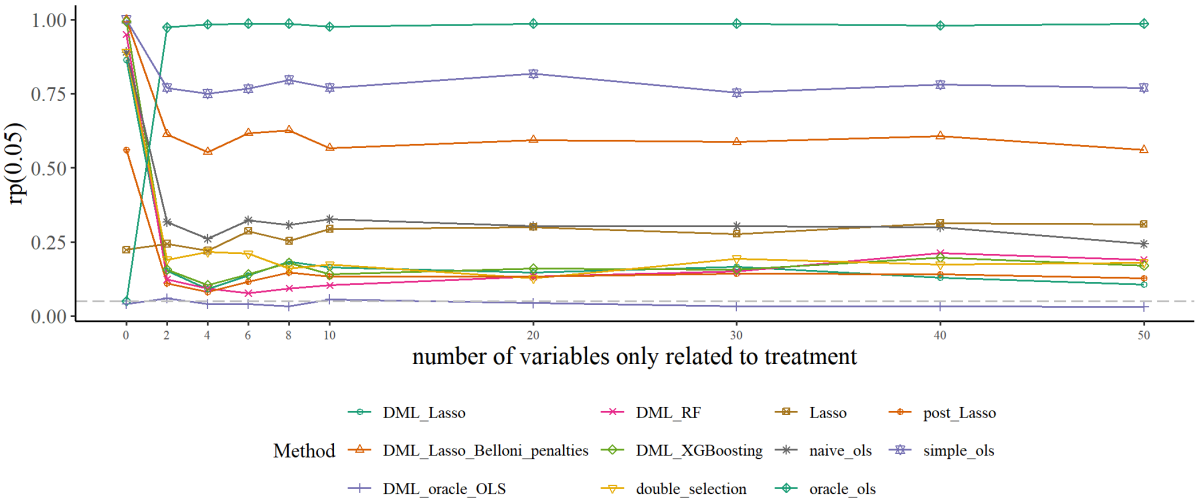

Figure 15: 95% rejection rate for the estimated bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of covariates only related to treatment

Figure 35 shows the standard deviation of the bias. We observe a slight increase in the standard deviation upon the addition of the first few treatment-only covariates, followed by a quick plateau. Among the models, post-LASSO exhibits the highest uncertainty in its bias and treatment effect estimates. Figure 15 illustrates the 95% coverage results. The models DML-LASSO, simple OLS, and DML-LASSO-Belloni-penalties demonstrate a higher rejection rate than the expected 5% when treatment-only covariates are included. This indicates a deviation from the prescribed coverage probability, suggesting that these models are less reliable in this context.

6.2.8 simulations 8.1-8.10: sensitivity to unobserved confounders

Figure 19 reports the sensitivity of mean bias for the 11 competing models to different numbers of omitted confounding variables. We observe significant differences in absolute mean bias performances for 0 and 1 omitted confounders, with a plateau at 2, indicating that the performance across all models stabilizes. The mean bias does not change substantially with the inclusion of any additional omitted confounder in the data-generating process (DGP). Feasible LASSO achieves near oracle OLS performance, while all other models perform relatively poorly.

Figures 17 and 18 presents the sensitivity of bias standard deviation for the 11 competing models to different numbers of omitted confounding variables, and the result for rejection rate. Again, we observe that just LASSO achieves near oracle performances in terms of rejection rate and low standard deviation. This indicates that LASSO-based methods, particularly feasible LASSO, are robust to the omission of confounders, maintaining low variability in bias and accurate rejection rates. All other models exhibit higher variability and poorer performance in the presence of omitted confounders, underscoring the importance of appropriate model selection in high-dimensional settings where confounders are likely to be missed.

6.2.9 simulations 9: including interaction and squared terms

Results for 13 different models in the replication simulation reduced to 20 covariates are presented in Table 6 in the appendix. These results are not significantly different from the default results. The results for the replication simulation with squared terms (where $p = 200$) are shown in Table 3. Most methods appear to perform better compared to Table1, where squared terms are not included. Notable exceptions are DML-LASSO-Belloni-penalties, simple-OLS, DML-LASSO, and feasible LASSO, which do not exhibit the same level of improvement. DML-RF, double selection, and Indirect Post-LASSO show improved performance in this case. This indicates that the gain from variables closer to the data-generating process (DGP) outweighs the extra cost induced by more noise variables. In other words, these methods are better able to perform variable selection.

In Table 4, default simulation 1 is replicated with the addition of squared and interaction terms. The best overall result occurs with the Belloni et al., 2011 double selection method in the situation where squared terms are included but interaction terms are not. This highlights the method's robustness in variable selection and its ability to handle the complexity introduced by non-linear relationships.

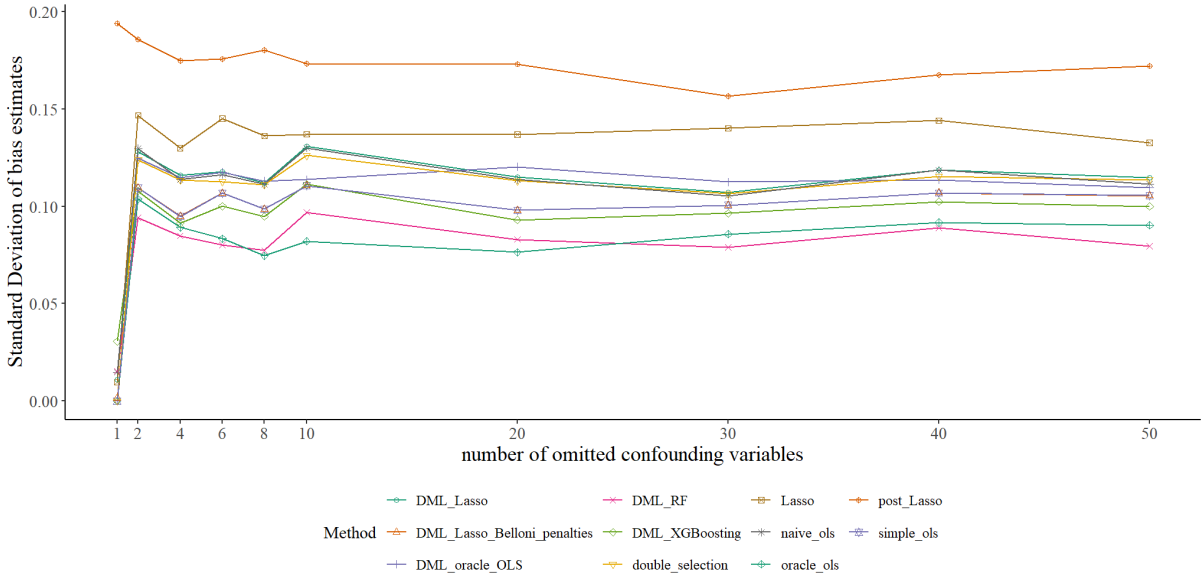

Figure 16: Standard deviation of the bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number omitted confounding variables

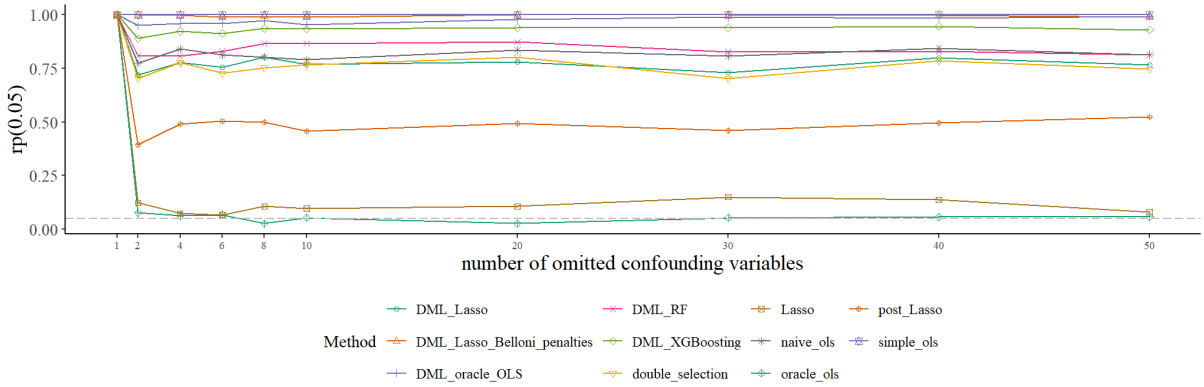

Figure 17: 95% rejection rate for the estimated bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number omitted confounding variables

Figure 18: 95% rejection rate for the estimated bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number omitted confounding variables

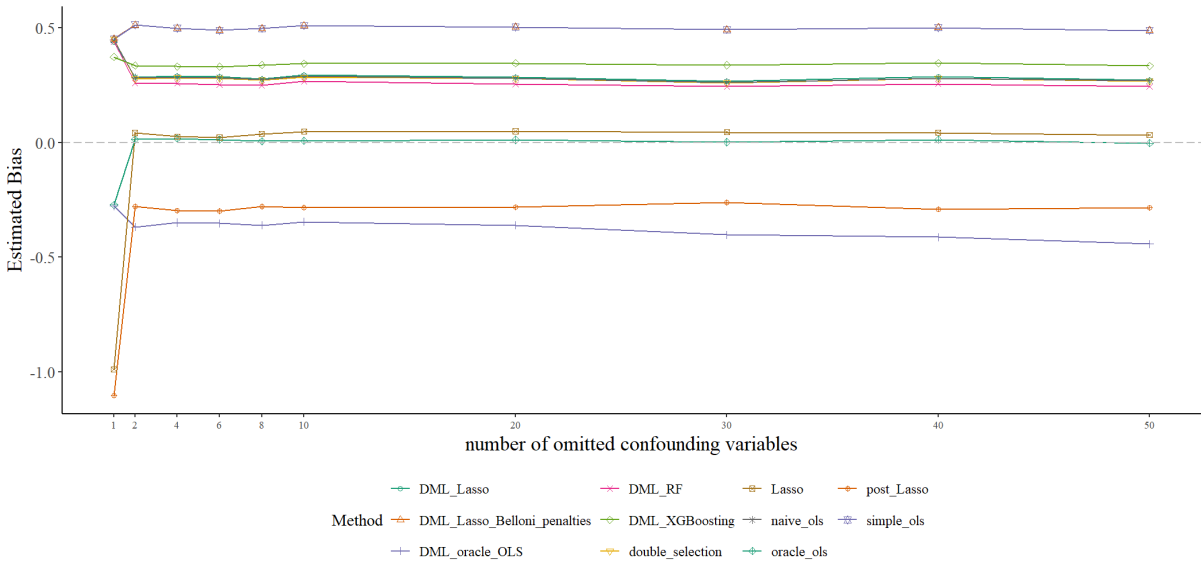

Figure 19: Mean bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number omitted confounding variables

Table 3: Results for 13 Different Models in replication simulation augmented with squared covariates

Model	Mean Bias	Standard Deviation	rp(0.05)
Oracle-OLS	-0.0007	0.1120	0.051
DML-RF	-0.0236	0.2559	0.044
Double selection Oracle	-0.0244	0.1122	0.072
double selection	-0.0444	0.1209	0.102
DML-oracle-OLS	-0.0514	0.1121	0.071
Post-Lasso	0.2326	0.2023	0.371
Indirect Post-Lasso	0.0214	0.1276	0.091
Lasso	0.4569	0.1443	0.838
DML-XGBoost	0.5693	0.1057	0.999
simple-OLS	0.7228	0.1022	1.000
DML-Lasso-Belloni-penalties	0.7228	0.1022	1.000
DML-Lasso	0.6607	0.0976	1.000

Note. naive-OLS is not reported as unfeasible (number of covariates bigger than number of observartions.

Table 4: Simulation results for default simulation 1 setup augmented with interaction and squared terms

Model	Mean Bias	Std Dev	rp(0.05)
Panel A: pc_sim_9.1.interaction.nosquare.sim1			
Oracle-OLS	-0.0029	0.0721	0.0633
DML-oracle-OLS	-0.0123	0.0764	0.0600
Post-Lasso	0.0043	0.1380	0.1800
DML-Lasso	0.0875	0.1024	0.1970
DML-Lasso-Belloni-penalties	0.1979	0.1674	0.5900
DML-XGBoost	0.4008	0.1157	0.9533
Double selection	0.0579	0.0922	0.1300
DML-RF	0.4153	0.0959	0.9700
Lasso	0.6007	0.1161	0.9933
Simple-OLS	0.6006	0.1160	1.0000
Panel B: pc_sim_9.3.nointeraction.square.sim1			
Oracle-OLS	-0.0029	0.0721	0.0633
DML-oracle-OLS	-0.0123	0.0764	0.0600
Post-Lasso	0.0003	0.1219	0.1700
DML-Lasso	0.0084	0.0932	0.0700
DML-Lasso-Belloni-penalties	0.2525	0.1618	0.7233
DML-XGBoost	0.4066	0.1155	0.9533
Double selection	-0.00002	0.0866	0.0700
DML-RF	0.2907	0.0949	0.8900
Lasso	0.6007	0.1161	0.9933
Simple-OLS	0.6006	0.1160	1.0000
Panel C: pc_sim_9.5.interaction.square.sim1			
Oracle-OLS	-0.0029	0.0721	0.0633
DML-oracle-OLS	-0.0123	0.0764	0.0600
Post-Lasso	0.0701	0.1282	0.2333
DML-Lasso	0.0021	0.0987	0.0967
DML-Lasso-Belloni-penalties	0.2445	0.1643	0.6667
DML-XGBoost	0.3985	0.1152	0.9500
Double selection	-0.0207	0.0856	0.0900
DML-RF	0.4115	0.0948	0.9700
Lasso	0.6007	0.1161	0.9933
Simple-OLS	0.6006	0.1160	1.0000

Note. naive-OLS is not reported as unfeasible (number of covariates bigger than number of observartions.

6.2.10 simulations 10: hyperparameter tuning of RF

We notice that the tuned DML-RF model on replication simulation performs worse than using the default hyperparameters. This is likely due to neyman hortogonality condition: if small perturbations in nuisance parameters (auxiliary ML parameters that are not of primary interest but are necessary for estimation) have minimal impact on the estimation of the parameter of interest.

7 Discussion and conclusion

The results of this thesis indicate that double machine learning (DML) methods, particularly DML-Random Forest and DML-LASO, offer superior performance in terms of bias and efficiency when compared to LASSO-based methods for treatment effect inference in high-dimensional, partially linear settings. The simulations reveal that DML methods are able to produce lower mean biases and better coverage probabilities, in most situation considered making them more reliable for practical applications then LASSO based ML methods for inference. However the best method is highly sensible to the simulation parameters, and thus the choiche of model(s) should be carefully considered based on the situation at hand.

Going back to the research question, the replication study shows that while Lasso-based methods are effective, DML methods generally exhibit lower bias and better efficiency. The DML-RF and DML-LASSO models, in particular, outperform other methods across various simulations. On the other hand it is also possible to point out a clear underperformer: the DML-LASSO-Belloni-penalties. The new hybrid method likely performs poorly due to a ML method for variable selection being used for a regression forecasting task.

The replication confirms the results of Belloni et al. (2011, Section 6.2) to a certain extent but highlights some discrepancies in bias rankings. The DML methods do not show better performances in this particular linear high-dimensional simulation setup.

The study provides practical guidelines indicating that DML-RF and DML-LASSO are preferable for high-dimensional datasets due to their robustness and lower bias in most simulation tested. However a strong limitation of this study is the reliance on simulated data, which may not capture all real-world complexities. Albeit providing the real DGP and treatment effect to know how biased the methods treatment effect estimates are, it could be that a real world application has a complex DGP that has not been considered here. For example the assumption of normality for the error terms could be relaxed as the methods in principle asymptotically allow for this (Belloni et al., 2013; Chernozhukov et al., 2018). Furthermore the sensitivity analysis has been carried out one dimension at a time. This does not fully allow to asses the relative bias performances of models when the default model is perturbed by more than one parameter changing. Futhure research with more computing power could focus on extending this thesis by doing grid-search amongst the simulation parameters. This way more accurate raccomandation for impact evaluation practitioners can be provided. Lastly the research lacks empirical applications to real-world datasets.

8 Acknowledgments

I would like to thank my supervisor, Sven Koobs, for his support and guidance throughout this research, including his availability for weekly office hours and his insightful feedback on my draft thesis and research proposal. I am also grateful to Erasmus University for providing access to the PC rooms, which were essential for carrying out the computationally intensive execution of this research. Additionally, I would like to express my gratitude to the academic authors that make their code and data available for replicability of their research. These replication files and packages have been helpful for the implementation of the thesis. Finally, I would like to thank my friends for their assistance in running my simulations code on their personal computers and for their availability in proofreading my work.

References

Au, T. C. (2018). Random forests, decision trees, and categorical predictors: The "absent levels" problem. *J. Mach. Learn. Res.*, *19*, Paper No. 45, 30.

Bach, P., Kurz, M. S., Chernozhukov, V., Spindler, M., & Klaassen, S. (2024). Doubleml: An object-oriented implementation of double machine learning in r. *Journal of Statistical Software*, *108*(3).

Belloni, A., Chen, D., Chernozhukov, V., & Hansen, C. (2012). Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, *80*(6), 2369–2429.

Belloni, A., Chernozhukov, V., & Hansen, C. (2011). Inference for high-dimensional sparse econometric models. *arXiv preprint*. https://arxiv.org/abs/1201.0220

Belloni, A., Chernozhukov, V., & Hansen, C. (2013). Inference on Treatment Effects after Selection among High-Dimensional Controls. *The Review of Economic Studies*, *81*(2), 608–650.

Belloni, A., Chernozhukov, V., & Hansen, C. (2014). High-dimensional methods and inference on structural and treatment effects. *Journal of Economic Perspectives*, *28*(2), 29–50.

Belloni, A., Chernozhukov, V., Hansen, C., & Damian, K. (2016). Inference in high-dimensional panel models with an application to gun control. *Journal of Business & Economic Statistics*, *34*(4), 590–605.

Bischl, B., Richter, J., Bossek, J., Horn, D., Thomas, J., & Lang, M. (2017). mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions. *arXiv preprint*. https://arxiv.org/abs/1703.03373

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

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Taylor & Francis.

Caron, a., Baio, G., & Manolopoulou, I. (2022). Shrinkage bayesian causal forests for heterogeneous treatment effects estimation. *Journal of Computational and Graphical Statistics*, *31*(4), 1202–1214.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

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

Chernozhukov, V., Hansen, C., & Spindler, M. (2016). High-dimensional metrics in r.

Donoho, D. L., et al. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, *1*(2000), 32.

Donohue, I., John J., & Levitt, S. D. (2001). The Impact of Legalized Abortion on Crime. *The Quarterly Journal of Economics*, *116*(2), 379–420.

Fradkov, A. L. (2020). Early history of machine learning [21st IFAC World Congress]. *IFAC-PapersOnLine*, *53*(2), 1385–1390.

Frisch, R., & Waugh, F. V. (1933). Partial time regressions as compared with individual trends. *Econometrica*, *1*(4), 387–401.

Fuhr, J., Berens, P., & Papies, D. (2024). Estimating causal effects with double machine learning–a method evaluation. *arXiv preprint*. https://arxiv.org/abs/2403.14385

Gertler, P. J., Martinez, S., Premand, P., Rawlings, L. B., & Vermeersch, C. M. (2016). *Impact evaluation in practice*. World Bank Publications.

Goodman-Bacon, A. (2021). Difference-in-differences with variation in treatment timing [Themed Issue: Treatment Effect 1]. *Journal of Econometrics*, *225*(2), 254–277.

Holten, S., Laarhoven, G., Polo, B., & Raina, M. (2024). *Mrf-arch: A machine learning approach to forecast usd/gbp tail risk in high dimensional settings [unpublished manuscript (seminar paper, available upon request)]*.

Johnson, M., Cao, J., & Kang, H. (2021). Detecting heterogeneous treatment effect with instrumental variables. https://arxiv.org/abs/1908.03652

Levitt, S. D. (1996). The effect of prison population size on crime rates: Evidence from prison overcrowding litigation. *The Quarterly Journal of Economics*, *111*(2), 319–351.

Lovell, M. C. (1963). Seasonal adjustment of economic time series and multiple regression analysis. *Journal of the American Statistical Association*, *58*(304), 993–1010.

MacKinnon, J. G., & White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, *29*(3), 305–325.

McConnell, K. J., & Lindner, S. (2019). Estimating treatment effects with machine learning. *Health Services Research*, *54*(6), 1273–1282.

Medeiros, M. C., Vasconcelos, G. F., Veiga, Á., & Zilberman, E. (2021). Forecasting inflation in a data-rich environment: The benefits of machine learning methods. *Journal of Business & Economic Statistics*, *39*(1), 98–119.

Qiu, M., Zigler, C., & Selin, N. E. (2022). Statistical and machine learning methods for evaluating trends in air quality under changing meteorological conditions. *Atmospheric Chemistry and Physics*, *22*(16), 10551–10566.

Staiger, D., & Stock, J. H. (1994). Instrumental variables regression with weak instruments. *National Bureau of Economic Research Technical Working Paper Series*. http://www.nber.org/papers/t01510

Tibshirani, R. (2018). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

A Programming code appendix

The `R` code is attached in a .zip file for replicability. To run it unzip it and open the README.txt file for instructions.

B Supplementary figures appendix

Table 5: Comparison of DML-RF Tuned and Non-Tuned Results for Sim 0

Method	**Mean Bias**	**Bias Std Dev**	**RP(0, 0.05)**
DML-RF Tuned	0.4105	0.1323	0.8580
DML-RF Non-Tuned	0.2990	0.1439	0.7470

Partial Linear Model Simulation Results

Estimator	Mean Bias	Std. Dev.	rp(0.05)
Lasso	0.644	0.093	1.000
Post-Lasso	0.415	0.209	0.877
Indirect Post-Lasso	0.0908	0.194	0.004
Double selection	-0.0041	0.111	0.054
Double selection Oracle	0.0001	0.110	0.051
Oracle	-0.0003	0.100	0.044

TABLE 4. Results are based on 1000 simulation replications of the partially linear model (6.53) where $p = 200$ and $n = 100$. We report mean bias (Mean Bias), standard deviation (Std. Dev.), and rejection frequency for 5% level tests (rp(.05)) for the four estimators described in Section 7.1.

Figure 20: Results to replicate. Adapted source: Belloni et al. (2011, Section 6.2).

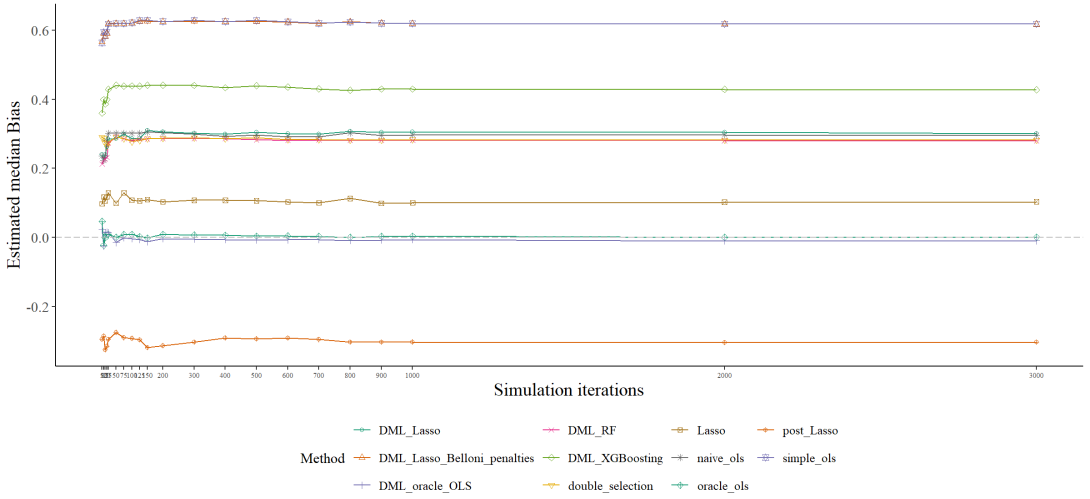

Figure 21: Median bias for 11 competing models (in legend) against number of MC simulation iterations

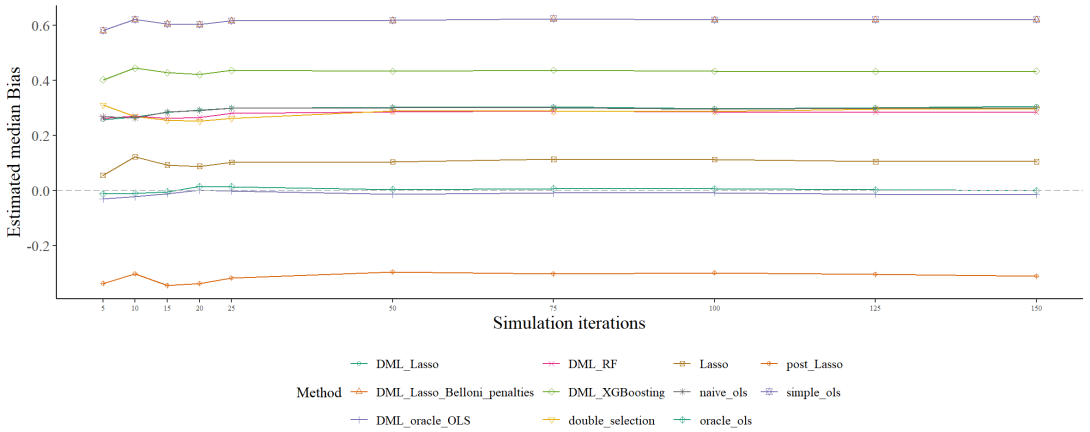

Figure 22: Median bias for 11 competing models (in legend) against number of MC simulation iterations. Selection of iterations up to 150.

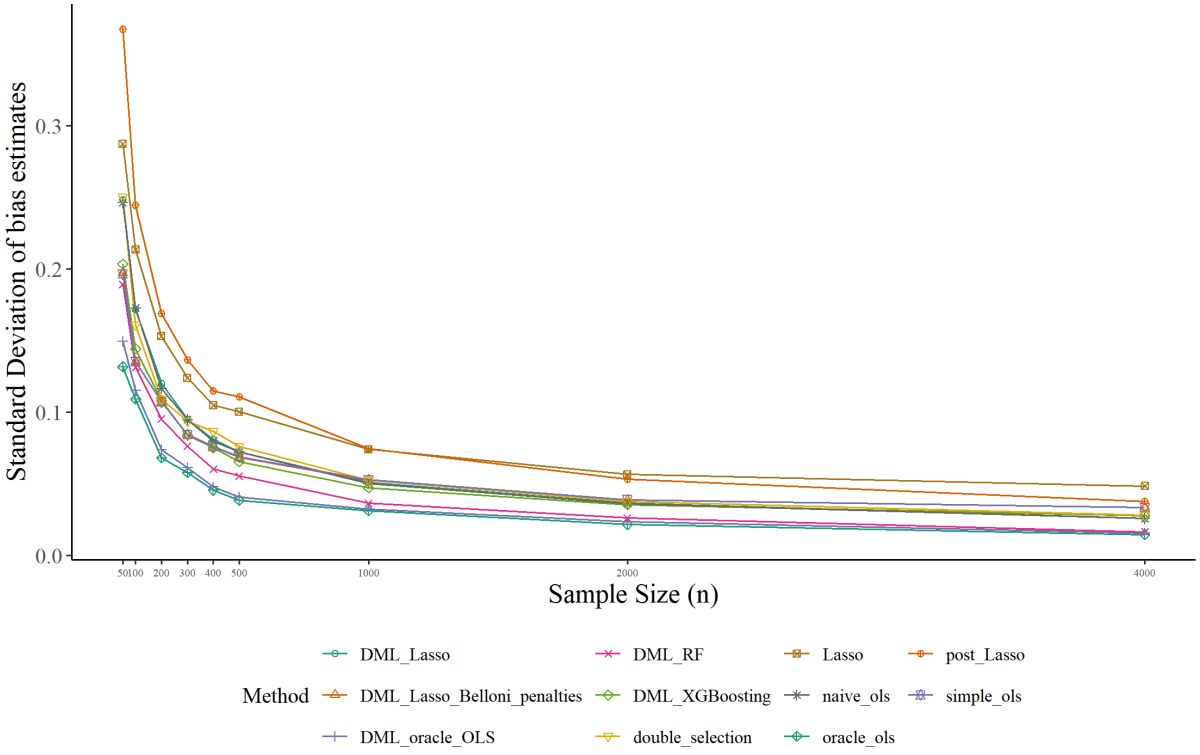

Figure 23: Standard deviation of the bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of observations

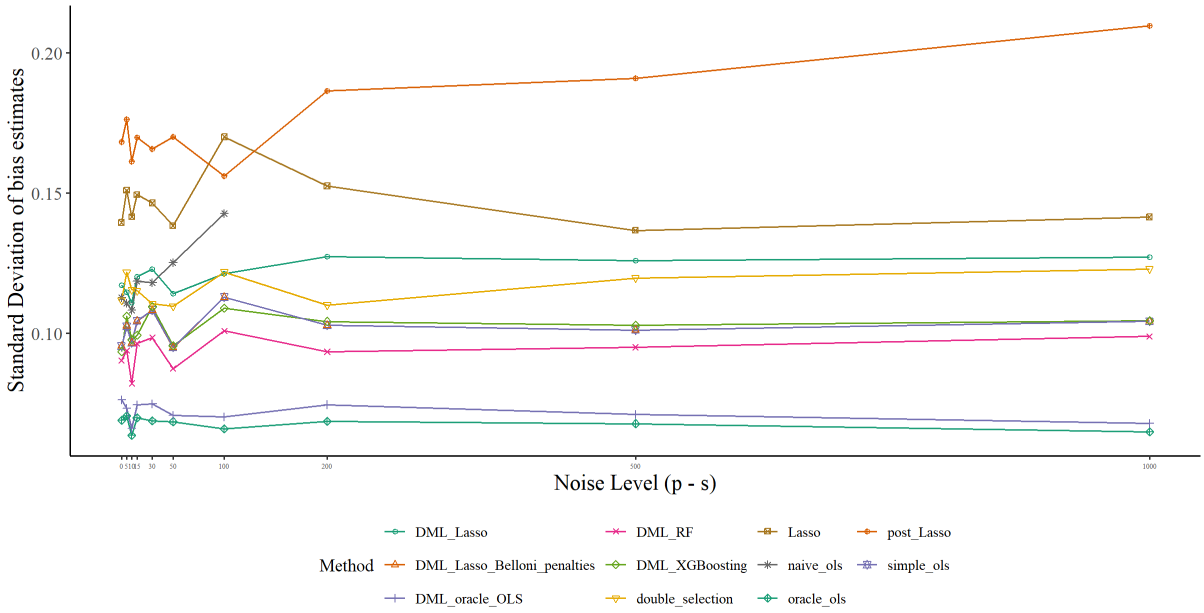

Figure 24: Standard deviation of the bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of observations

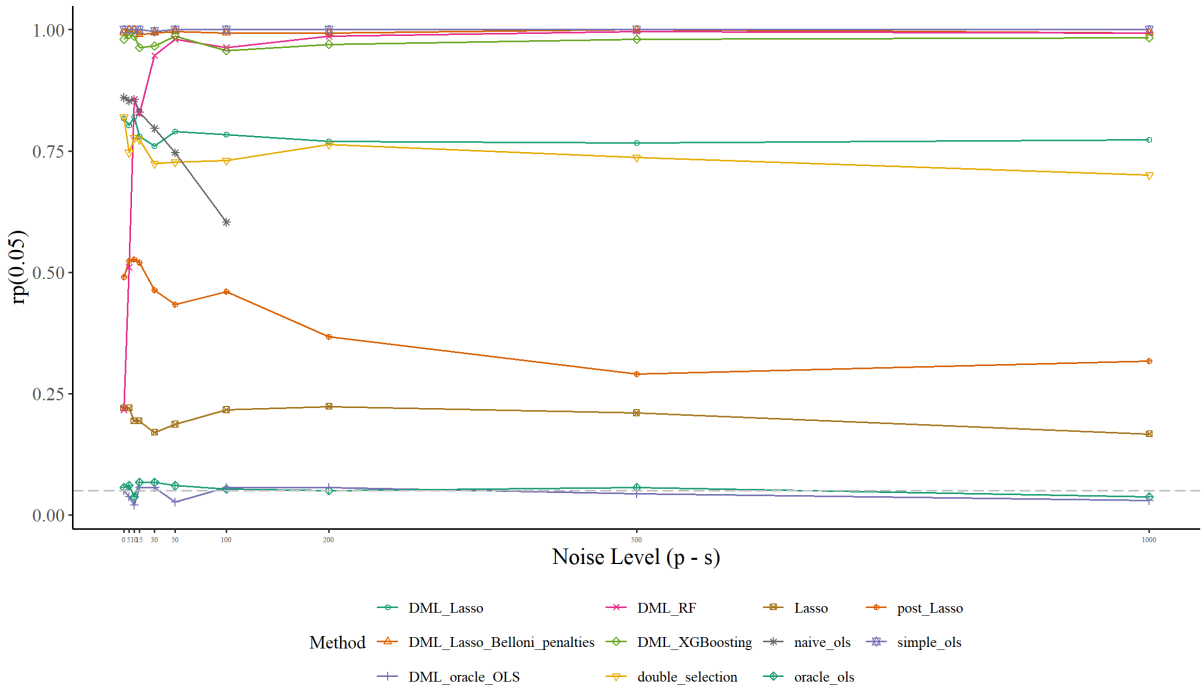

Figure 25: 95% rejection rate for the estimated bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of noise covariates

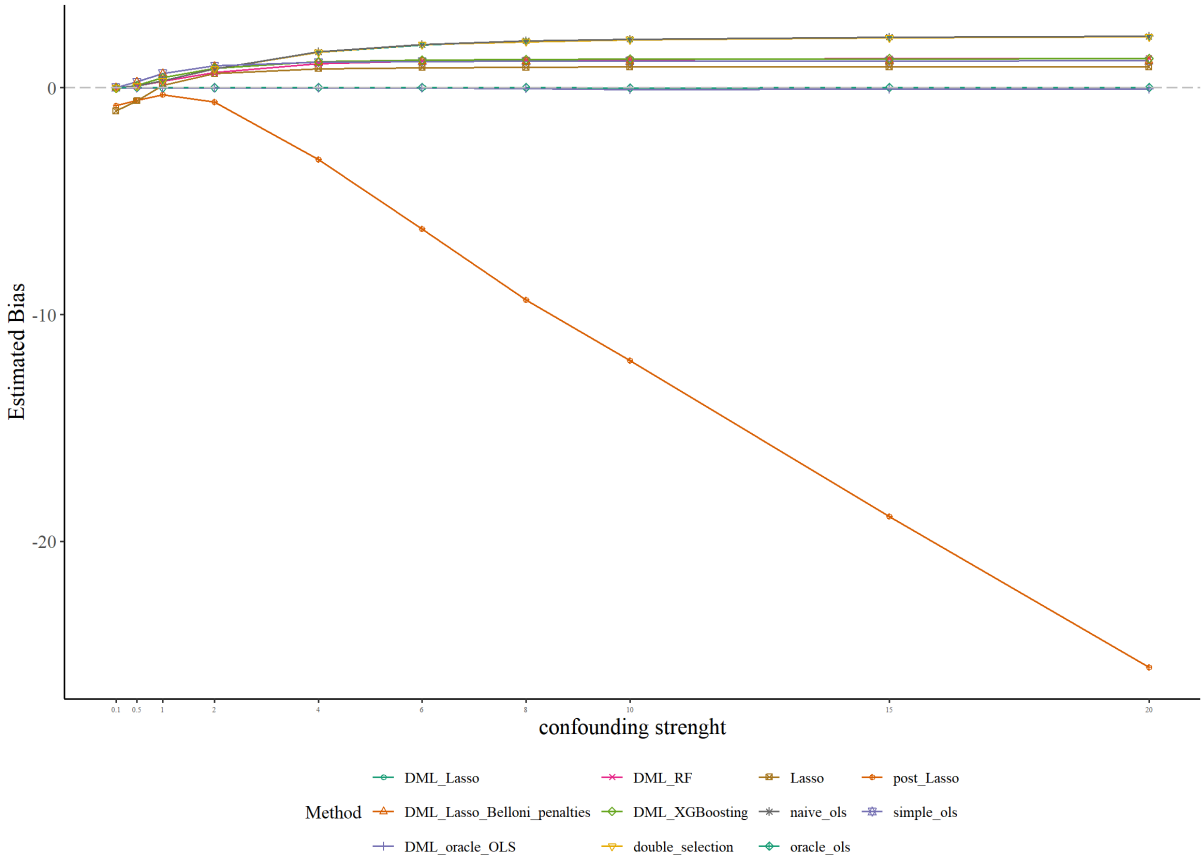

Figure 26: Mean bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different confounding strength multipliers

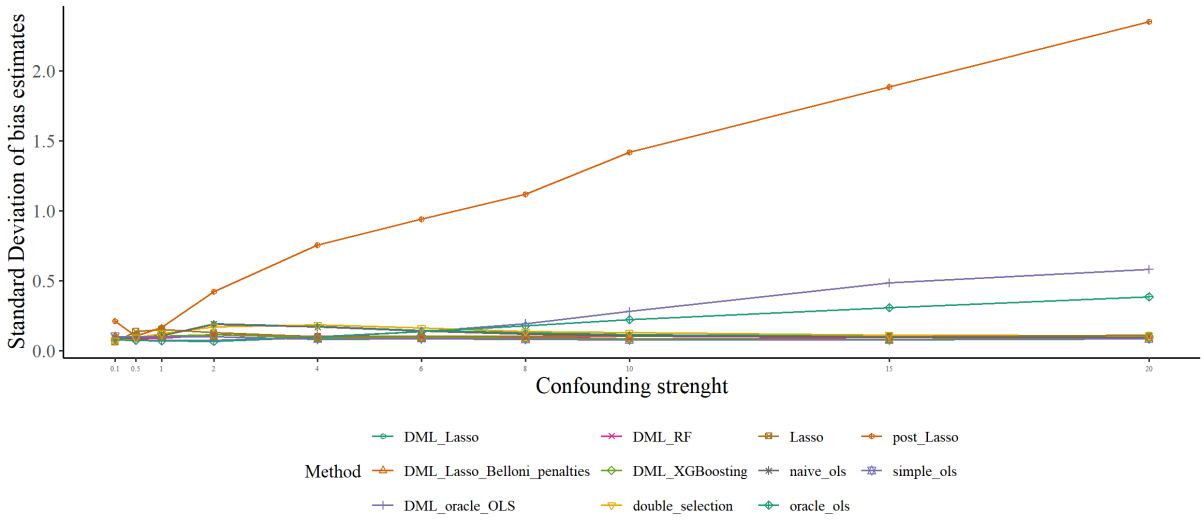

Figure 27: Standard deviation of the bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different confounding strength multipliers

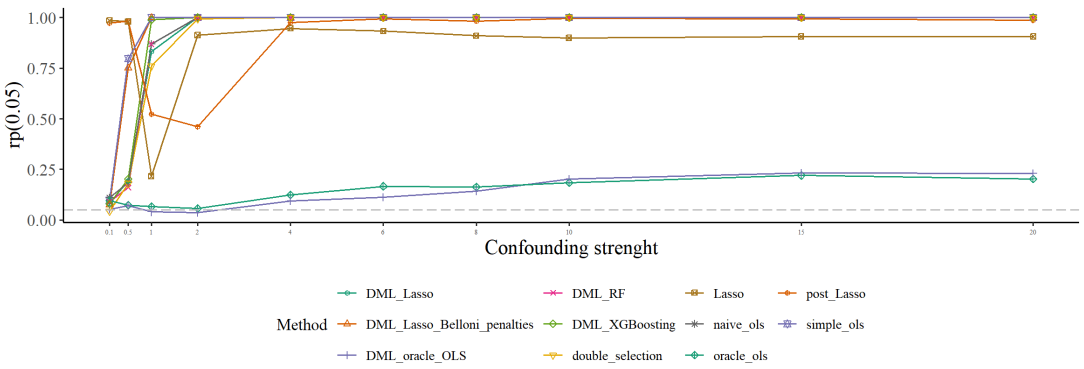

Figure 28: 95% rejection rate for the estimated bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different confounding strength multipliers

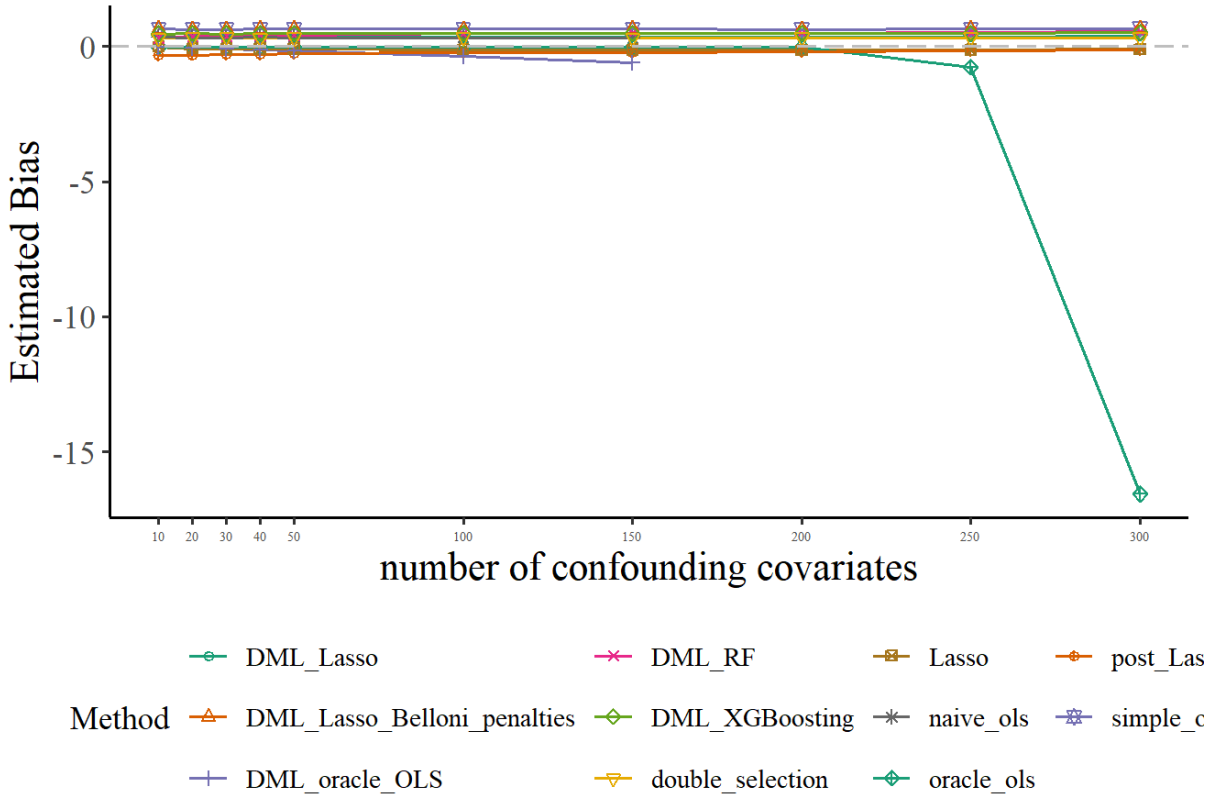

Figure 29: Mean bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of confounding covariates

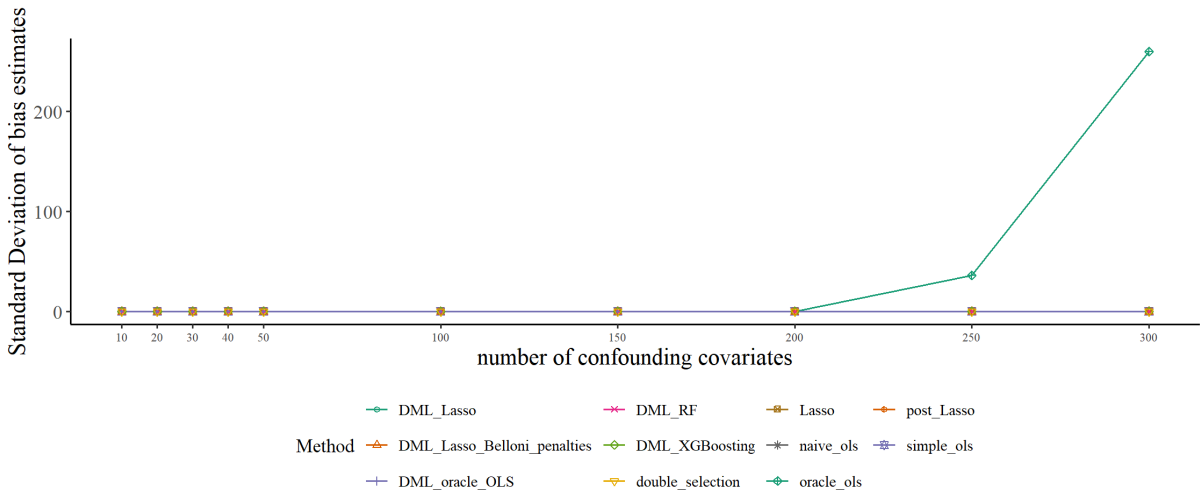

Figure 30: Standard deviation of the bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of confounding covariates

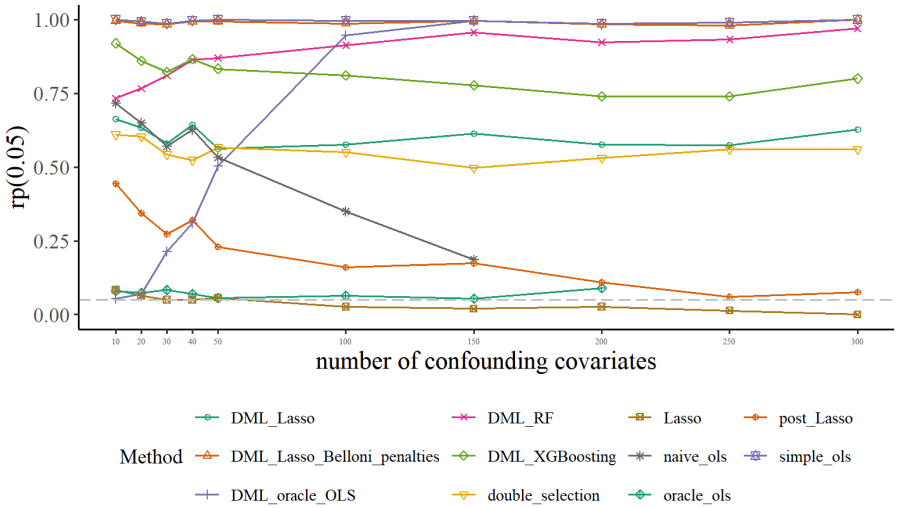

Figure 31: 95% rejection rate for the estimated bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of confounding covariates

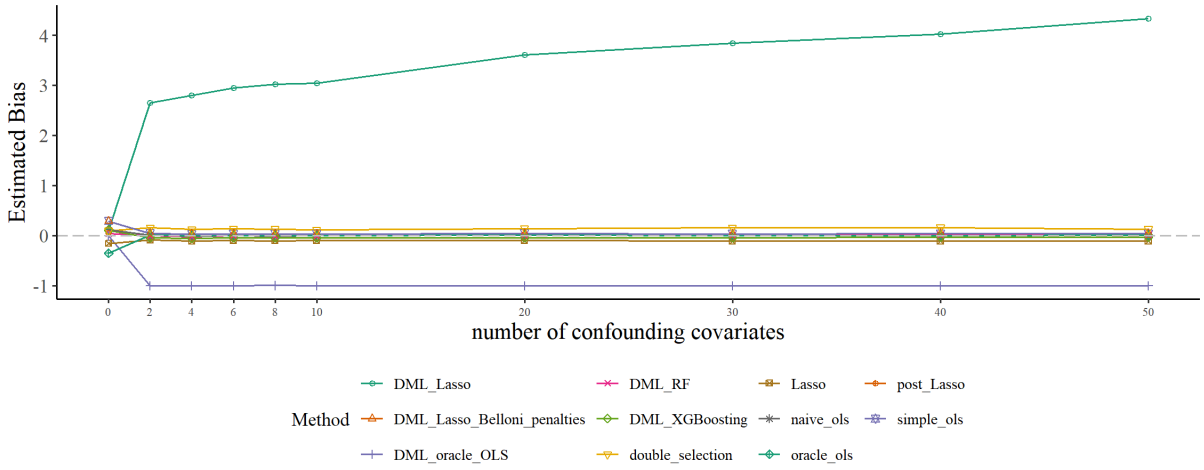

Figure 32: Mean bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of covariates only related to outcome

Note: there is a mistake in the labelling of the axis, It should be:
number of covariates only related to outcome

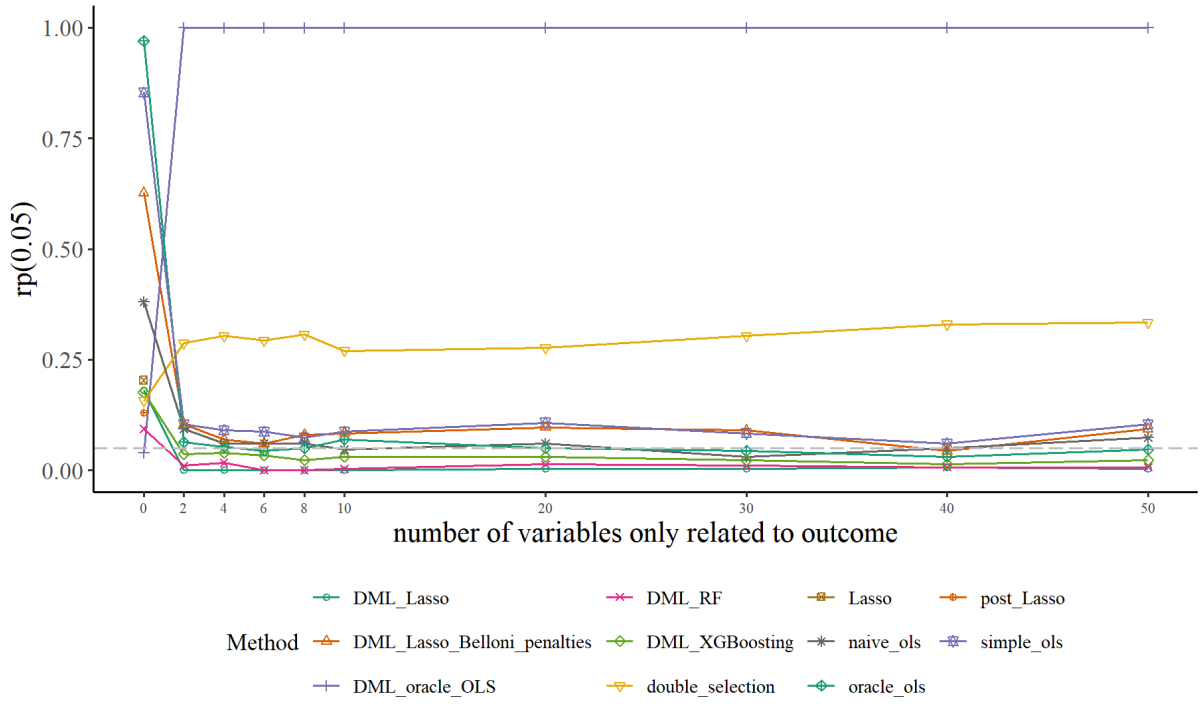

Figure 33: 95% rejection rate for the estimated bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of covariates only related to outcome

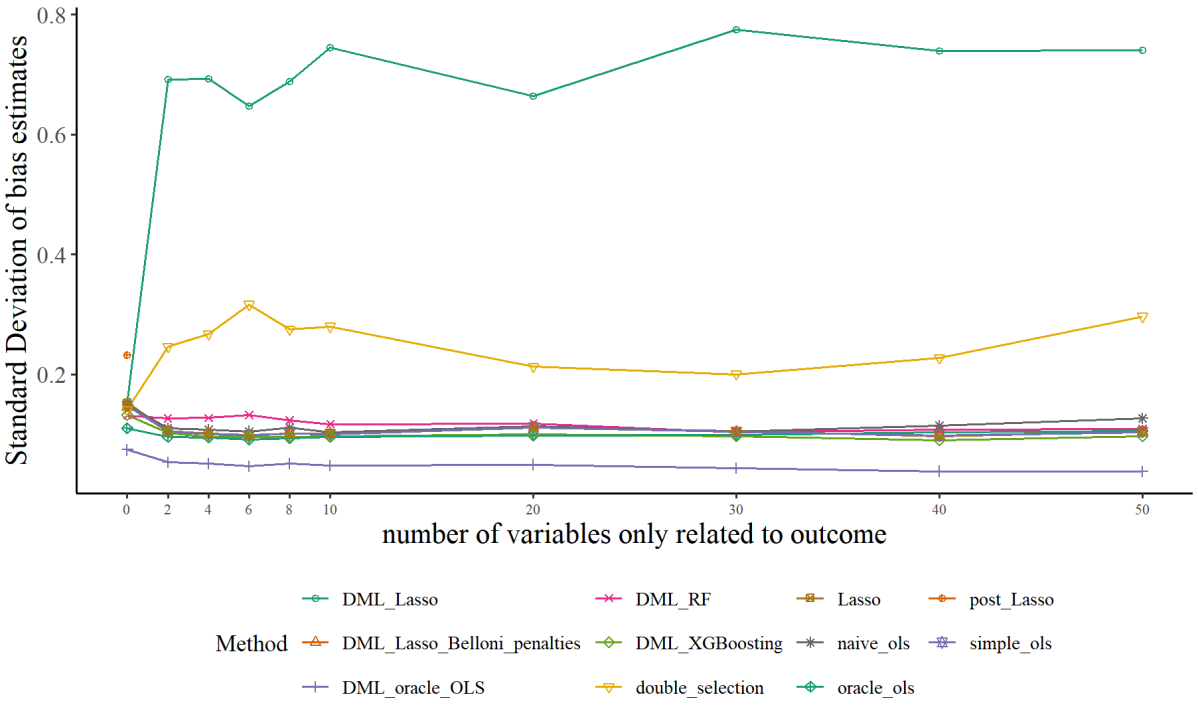

Figure 34: Standard deviation of the bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of covariates only related to outcome

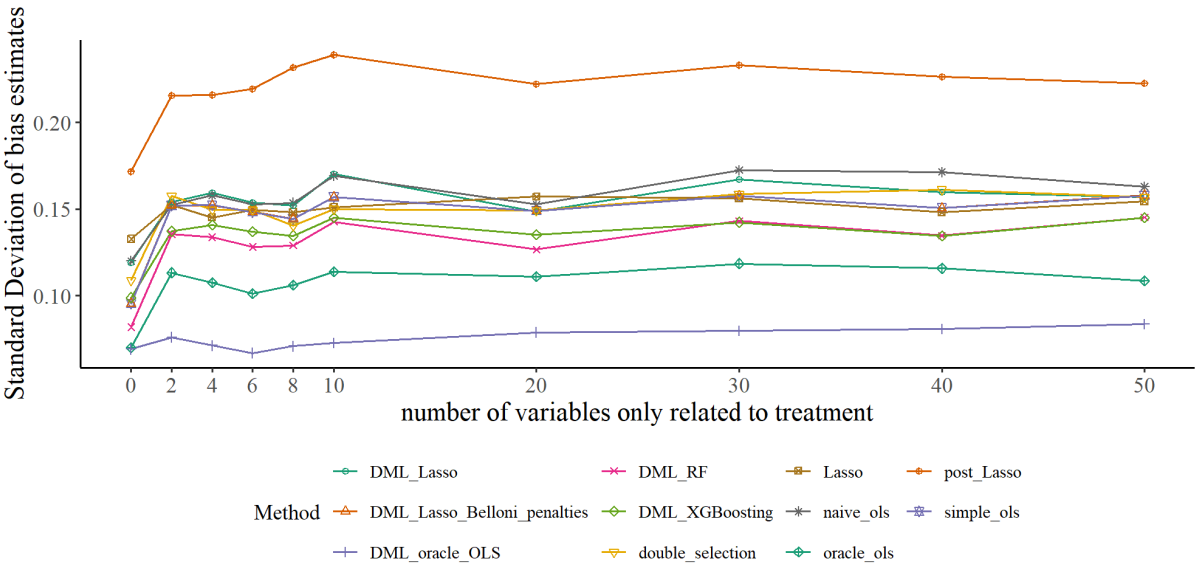

Figure 35: Standard deviation of the bias for 11 competing models (in legend), over 300 Monte Carlo iterations, for different number of covariates only related to treatment

C Supplementary tables appendix

Table 6: Results for 13 different models in replication simulation with 20 covariates

no interaction terms, no squares (benchmark $p = 20$)			
Model	Mean Bias	Standard Deviation	rp(0.05)
DML-Lasso	-0.0001	0.1134	0.070
naive-OLS	0.0004	0.1153	0.056
Oracle-OLS	-0.0014	0.1104	0.049
Indirect Post-Lasso	-0.0247	0.1559	0.028
Double selection Oracle	-0.0248	0.1107	0.065
double selection	-0.0268	0.1113	0.066
DML-oracle-OLS	-0.0505	0.1097	0.061
Post-Lasso	0.0372	0.1623	0.148
DML-RF	0.4038	0.1027	0.938
Lasso	0.5172	0.1455	0.971
DML-XGBoost	0.5707	0.1042	0.999
simple-OLS	0.7176	0.1078	1.000
DML-Lasso-Belloni-penalties	0.7176	0.1080	1.000
interaction terms, squares			
Model	Mean Bias	Standard Deviation	rp(0.05)
Oracle-OLS	0.0013	0.1068	0.049
Indirect Post-Lasso	-0.0162	0.2235	0.039
Double selection Oracle	-0.0222	0.1059	0.062
double selection	-0.0502	0.1158	0.089
DML-oracle-OLS	-0.0497	0.1064	0.067
DML-Lasso	0.0166	0.1251	0.083
Post-Lasso	0.1529	0.1983	0.261
DML-RF	0.6269	0.1019	1.000
Lasso	0.4684	0.1493	0.865
DML-XGBoost	0.5708	0.1066	0.998
simple-OLS	0.7195	0.1076	1.000
DML-Lasso-Belloni-penalties	0.7197	0.1075	1.000
interaction terms, no squares			
Model	Mean Bias	Standard Deviation	rp(0.05)
Oracle-OLS	-0.0014	0.1104	0.049
Indirect Post-Lasso	-0.0247	0.1559	0.028
Double selection Oracle	-0.0248	0.1107	0.065
double selection	-0.0268	0.1113	0.066
DML-oracle-OLS	-0.0505	0.1097	0.061
DML-Lasso	0.0114	0.1220	0.082
Post-Lasso	0.1534	0.1852	0.252
DML-RF	0.6209	0.1039	1.000
Lasso	0.4691	0.1496	0.879
DML-XGBoost	0.5683	0.1071	0.999
simple-OLS	0.7176	0.1078	1.000
DML-Lasso-Belloni-penalties	0.7176	0.1080	1.000

Note. naive-OLS is not reported in panel 2 and 3 as unfeasible (number of covariates bigger than number of observations).