Model averaging and double machine learning

EEA/ESEM

Barcelona School of Economics, Barcelona
June 29, 2023

Achim Ahrens
ETH Zürich

Christian B. Hansen
University of Chicago

Mark E. Schaffer
Heriot-Watt University, IZA

Thomas Wiemann
University of Chicago

Introduction

Machine learning (ML) is increasingly popular to aid causal effect estimation – For example:

- ▷ Post-double-selection (PDS) lasso (Belloni et al., 2014)
- ▷ Double/debiased machine learning (DDML) (Chernozhukov et al., 2018)

Key theoretical benefit:

- ▷ Statistical inference despite high-dimensional confounding factors

Introduction

Machine learning (ML) is increasingly popular to aid causal effect estimation – For example:

- Post-double-selection (PDS) lasso (Belloni et al., 2014)
- Double/debiased machine learning (DDML) (Chernozhukov et al., 2018)

Key theoretical benefit:

- Statistical inference despite high-dimensional confounding factors

Recent literature ***raises concerns*** about practical advantages of ML:

- Goller et al. (2020): Random forests + matching "might lead to misleading results."
- Wüthrich and Zhu (2021): Lasso selection of controls can introduce OVB in small samples.
- Angrist and Frandsen (2022): "ML seems ill-suited to IV applications in labor economics."

Introduction

In this paper, we...

- revisit some of the concerns expressed in the literature.
- discuss pairing DDML with stacking (a.k.a. model averaging, 'super learning').
- introduce ***short-stacking***, which substantially reduces the computational burden of DDML+stacking; and ***pooled stacking***.
- formulate recommended practices & provide complementary Stata and R software to implement our recommendations.

Motivation

Assumption 1 (Partially Linear Regression; PLR)

Let (Y, D, X, U) be a random vector w/ distribution characterized by

$$Y = \tau D + m(X) + U, \quad E[U|D, X] = 0, \tag{1}$$

Motivation

Assumption 1 (Partially Linear Regression; PLR)

Let (Y, D, X, U) be a random vector w/ distribution characterized by

$$Y = \tau D + m(X) + U, \quad E[U|D, X] = 0, \tag{1}$$

Typical approach: we assume $m(X) = X'\beta$ and apply least squares.

Why use something else than least squares?

- ▷ We have many controls (relative to the sample size), but do not know which to include.
- ▷ We have unknown non-linear structures.

Assumption 1 (Partially Linear Regression; PLR)

Let (Y, D, X, U) be a random vector w/ distribution characterized by

$$Y = \tau D + m(X) + U, \quad E[U|D, X] = 0, \tag{2}$$

where $\tau \in \mathbb{R}$ and $m : \operatorname{supp} X \to \mathbb{R}$.

Constructed moment condition gives familiar expression:

$$E\left[(Y - E[Y|X] - \tau\,(D - E[D|X]))\,(D - E[D|X])\right] = 0$$

$$\Rightarrow \qquad \tau = \frac{E\left[(Y - E[Y|X])\,(D - E[D|X])\right]}{E\left[(D - E[D|X])^2\right]}.$$

Motivation

Assumption 1 (Partially Linear Regression; PLR)

Let (Y, D, X, U) be a random vector w/ distribution characterized by

$$Y = \tau D + m(X) + U, \quad E[U|D, X] = 0, \tag{2}$$

where $\tau \in \mathbb{R}$ and $m : \operatorname{supp} X \to \mathbb{R}$.

Constructed moment condition gives familiar expression:

$$\begin{aligned} & E\left[(Y - E[Y|X] - \tau(D - E[D|X]))(D - E[D|X])\right] = 0 \\ \Rightarrow \quad & \tau = \frac{E\left[(Y - E[Y|X])(D - E[D|X])\right]}{E\left[(D - E[D|X])^2\right]}. \end{aligned}$$

Idea: Use ML to estimate conditional expectation functions (CEFs).

Double/Debiased Machine Learning

However, plugging in ML estimates of CEFs generally induces an ***over-fitting bias***.

Double/Debiased Machine Learning (Chernozhukov et al., 2018)

- ▷ relies on ***sample-splitting/cross-fitting*** and ***Neyman-orthogonal moment conditions***,
- ▷ can be combined with a ***general class of ML methods***,
- ▷ requires only relatively mild rate requirements for asymptotic normality,
- ▷ can be used to estimate structural parameters in various models (beyond the partially linear model).

The cross-fitting algorithm

1. splits the sample I randomly into K folds denoted $I_1, \ldots, I_K$,
2. fits CEF estimators iteratively on the sample excluding the hold-out fold, i.e., $I \setminus I_k$,
3. calculates the out-of-sample predicted values for the hold-out fold I_k,
4. and uses these *'cross-fitted'* predicted values to estimate the structural parameters on the full sample I.

The choice of machine learner

Which machine learner should we use?

Which machine learner performs best in a particular application depends crucially on ***match quality of machine learner & structure of the DGP***.

- There is no general answer to the question of whether lasso or random forests will ‘work’ or will not ‘work’ in a given application.
- No-free lunch theorem in machine learning (Wolpert, 1996; Wolpert and Macready, 1997).
- Machine learners require ‘tuning’ (e.g., tree-depth, learning rate).

The choice of machine learner

For example, ***the lasso*** has become a popular tool in empirical economics.

- ▷ intuitive assumption of (approximate) sparsity
- ▷ computationally relatively cheap
- ▷ linearity has its advantages (e.g. extension to panel data; Belloni et al., 2016)

The choice of machine learner

For example, ***the lasso*** has become a popular tool in empirical economics.

- ▷ intuitive assumption of (approximate) sparsity
- ▷ computationally relatively cheap
- ▷ linearity has its advantages (e.g. extension to panel data; Belloni et al., 2016)

But there are also drawbacks:

- ▷ What if the ***sparsity assumption*** is not plausible?
  $\rightarrow$ "Illusion of Sparsity" (Giannone et al., 2021)
- ▷ There is a wide set of machine learners at disposal—lasso might not be the best choice for a particular application.

DDML+stacking

Stacking allows for ***combining multiple*** CEF estimators.

- ▷ constructs weighted average of ‘candidate’ learners
- ▷ performs asymptotically ***at least as well as the best-performing candidate learner*** if number of candidates grows at most at polynomial rate (der Laan et al., 2007; Polley et al., 2011)
- ▷ Van der Laan et al. (2011) advocate for stacking (“super learning”) for Targeted MLE

DDML+stacking

Stacking allows for ***combining multiple*** CEF estimators.

- ▷ constructs weighted average of ‘candidate’ learners
- ▷ performs asymptotically ***at least as well as the best-performing candidate learner*** if number of candidates grows at most at polynomial rate (der Laan et al., 2007; Polley et al., 2011)
- ▷ Van der Laan et al. (2011) advocate for stacking (“super learning”) for Targeted MLE

We illustrate: Stacking safeguards against ill-chosen/poorly tuned learners ***provided a generous and diverse*** set of base learners is included.

DDML+stacking

In each cross-fitting step $k = 1, \dots, K$,

$$\min_{w_{k,1},\dots,w_{k,J}} \sum_{i \in T_k} \left(Y_i - \sum_{j=1}^{J} w_{k,j} \hat{\ell}^{(j)}_{T^c_{k,v(i)}}(\boldsymbol{X}_i) \right)^2 \qquad \text{s.t. } w_{k,j} \geq 0, \; \sum_{j=1}^{J} |w_{k,j}| = 1.$$

where $\hat{\ell}^{(j)}_{T^c_{k,v(i)}}(\boldsymbol{X}_i) \equiv$ cross-validated predicted values, $J \equiv$ number of candidate learners, $V \equiv \#$ CV folds.

Final stacking estimator: $\hat{\ell} = \sum_{j=1}^{J} \hat{w}_j \hat{\ell}_j$.

Other options: single-best learner, unconstrained OLS, unweighted average, etc.

Result of der Laan et al. (2007) does not require non-negativity or sum-to-one constraint.

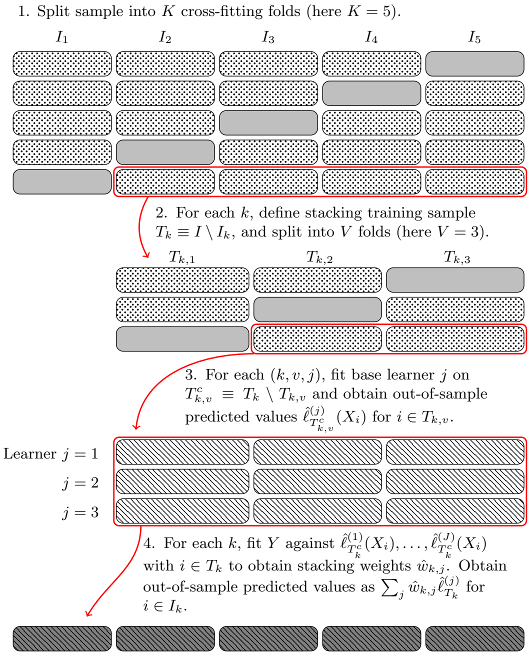


Figure 1: Cross-fitting and stacking. Example: estimation of $\ell_0 = E[Y|X]$.

DDML+stacking

Two drawbacks of pairing DDML with (regular) stacking:

- ▷ computational complexity: $K \times V \times J$ learners are fit where $K =$ cross-fitting folds, $V =$ cross-validation folds, $J =$ number of candidate learners
- ▷ possibly sub-optimal performance in small samples given that learners are fit on $(K-1)(V-1)/(KV)\%$ of the sample

DDML+stacking

Two drawbacks of pairing DDML with (regular) stacking:

- ▷ computational complexity: $K \times V \times J$ learners are fit where $K =$ cross-fitting folds, $V =$ cross-validation folds, $J =$ number of candidate learners
- ▷ possibly sub-optimal performance in small samples given that learners are fit on $(K-1)(V-1)/(KV)\%$ of the sample

Short-stacking takes a short-cut by training the final learner on the cross-fitted values using the full sample. The objective function becomes:

$$\min_{w_1,\ldots,w_J} \sum_{i=1}^{n} \left(Y_i - \sum_{j=1}^{J} w_j \hat{\ell}^{(j)}_{I^c_{k(i)}}(\boldsymbol{X}_i) \right)^2 \qquad \text{s.t. } w_j \geq 0,\ \sum_j |w_j| = 1$$

where w_j are the short-stacking weights. Cross-fitting serves a ***double purpose***: addressing own-observation bias and yielding out-of-sample predicted values used for estimating weights.

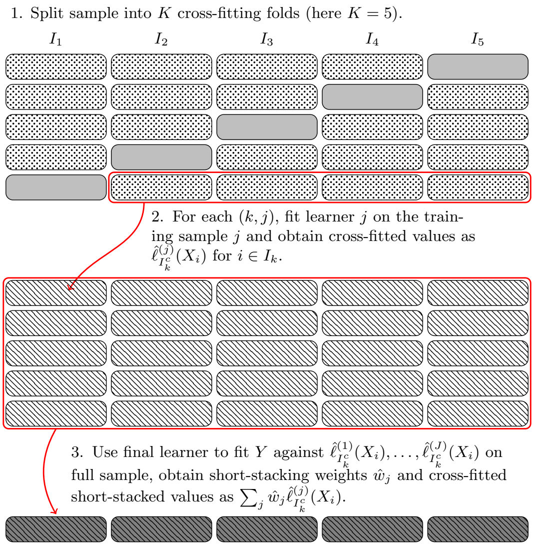


Figure 2: Cross-fitting & *short*-stacking. Example: estimation of $\ell_0 = E[Y|X]$.

Simulation I: Advantages of DDML+Stacking

Calibrated simulation based on Poterba et al. (1995), who estimate the causal effect of 401(k) eligibility on wealth.

- outcome: log wealth
- treatment: 401(k) eligibility
- controls: age, income, education in years, family size, two-earner status, home ownership, and participation in two alternative pension schemes.
- $N = 9,915$

Simulation design: Reconstruct CEFs with either OLS (*linear DGP*) or gradient boosting (*nonlinear DGP*) to reinforce the linear/non-linear signal in the data. Simulation design

- 1,000 bootstrap draws
- 10 candidate learners including OLS, CV-lasso/ridge, random forests, gradient boosting, feed-forward neural net Details

Simulation I: Advantages of DDML+Stacking

Table 1: Bias and Coverage Rates in the ***Linear DGP*** Table notes More results

	$n_b = 9{,}915$			$n_b = 99{,}150$		
Panel (A): Linear DGP	Bias	MAB	Rate	Bias	MAB	Rate
Full sample:						
OLS	−24.8	820.2	0.95	−1.9	269.1	0.95
PDS-Lasso	−25.4	821.2	0.95	0.6	269.4	0.95
DDML methods:						
Base learners						
OLS	−30.5	826.8	0.95	−2.3	270.5	0.95
Lasso with CV (2nd order poly)	−28.5	830.2	0.96	−1.5	272.6	0.95
Ridge with CV (2nd order poly)	−25.5	821.1	0.95	−1.9	275.9	0.95
Lasso with CV (10th order poly)	187.4	1047.6	0.95	61.4	281.7	0.94
Ridge with CV (10th order poly)	1069.3	1221.0	0.94	38.1	273.4	0.94
Random forest (low regularization)	−196.5	982.2	0.91	−33.3	356.7	0.87
Random forest (high regularization)	−28.1	853.1	0.95	−22.5	288.7	0.94
Gradient boosting (low regularization)	−82.2	825.0	0.95	−19.4	270.9	0.95
Gradient boosting (high regularization)	28.6	819.6	0.96	71.4	279.8	0.94
Neural net	309.2	866.0	0.94	17.4	288.1	0.94
Meta learners						
Stacking: CLS	8.8	829.3	0.95	−1.5	274.4	0.95
Stacking: Single-best	−28.6	823.3	0.96	−3.5	272.1	0.95
Short-stacking: CLS	−23.8	817.6	0.96	−1.5	274.3	0.95
Short-stacking: Single-best	−20.7	826.0	0.96	−3.4	272.4	0.95

Simulation I: Advantages of DDML+Stacking

Table 2: Bias and Coverage Rates in the ***Non-Linear DGP*** Table notes More results

	n_b = 9,915			n_b = 99,150		
Panel (B): Non-Linear DGP	Bias	MAB	Rate	Bias	MAB	Rate
Full sample:						
OLS	−2587.3	2642.4	0.58	−2644.9	2640.5	0.
PDS-Lasso	−2598.4	2661.3	0.57	−2644.1	2638.4	0.
DDML methods:						
Base learners						
OLS	−2617.9	2622.6	0.58	−2647.2	2645.8	0.
Lasso with CV (2nd order poly)	746.2	1105.5	0.89	703.5	714.6	0.61
Ridge with CV (2nd order poly)	801.9	1143.6	0.89	714.3	725.3	0.60
Lasso with CV (10th order poly)	−4684.7	2111.1	0.90	−2.1	284.5	0.94
Ridge with CV (10th order poly)	−3070.8	2499.6	0.87	−1.9	287.5	0.95
Random forest (low regularization)	−64.0	1065.0	0.90	−43.4	331.8	0.87
Random forest (high regularization)	−133.0	932.3	0.94	−18.5	272.8	0.94
Gradient boosting (low regularization)	52.4	924.3	0.93	13.6	267.7	0.95
Gradient boosting (high regularization)	199.8	895.4	0.94	182.3	319.3	0.93
Neural net	−620.3	1103.6	0.91	−142.9	292.0	0.92
Meta learners						
Stacking: CLS	226.8	1129.3	0.93	24.9	262.9	0.95
Stacking: Single-best	179.2	1016.2	0.92	15.0	266.4	0.95
Short-stacking: CLS	221.9	897.3	0.93	21.9	261.7	0.95
Short-stacking: Single-best	116.1	900.1	0.94	15.0	266.4	0.95

Simulation I: Advantages of DDML+Stacking

Table 3: Average stacking weights

	Stacking		Short-stacking	
Panel (A): Linear DGP	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$
OLS	0.680	0.494	0.677	0.485
Lasso with CV (2nd order poly)	0.096	0.138	0.114	0.135
Ridge with CV (2nd order poly)	0.068	0.058	0.080	0.069
Lasso with CV (10th order poly)	0.030	0.074	0.022	0.076
Ridge with CV (10th order poly)	0.028	0.043	0.027	0.055
Random forest (low regularization)	0.013	0.011	0.008	0.007
Random forest (high regularization)	0.018	0.028	0.012	0.024
Gradient boosting (low regularization)	0.028	0.045	0.025	0.043
Gradient boosting (high regularization)	0.020	0.062	0.019	0.061
Neural net	0.019	0.047	0.016	0.044
Panel (B): Non-Linear DGP	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$
OLS	0.011	0.015	0.004	0.010
Lasso with CV (2nd order poly)	0.036	0.059	0.016	0.038
Ridge with CV (2nd order poly)	0.153	0.231	0.123	0.230
Lasso with CV (10th order poly)	0.060	0.076	0.052	0.063
Ridge with CV (10th order poly)	0.074	0.061	0.061	0.056
Random forest (low regularization)	0.042	0.012	0.043	0.006
Random forest (high regularization)	0.021	0.076	0.014	0.063
Gradient boosting (low regularization)	0.523	0.229	0.615	0.346
Gradient boosting (high regularization)	0.014	0.192	0.005	0.139
Neural net	0.067	0.049	0.066	0.050

Simulation I: Advantages of DDML+Stacking

As expected, OLS performs best in the fully linear setting and DDML+GB performs best in the when the nuisance function is generated by gradient boosting.

Simulation I: Advantages of DDML+Stacking

As expected, OLS performs best in the fully linear setting and DDML+GB performs best in the when the nuisance function is generated by gradient boosting.

In practice, researchers rarely know the functional structure in economic applications.

- ▷ Stacking & short-stacking assign high weights to the data-generating learner.
- ▷ Stacking reduces the ***burden of choice*** the researcher faces by allowing for the simultaneous consideration of multiple estimators.
- ▷ DDML paired with short-stacking performs very similar to DDML w/ regular stacking, despite lower computational burden (speed gain by factor $1/V$). Computational time

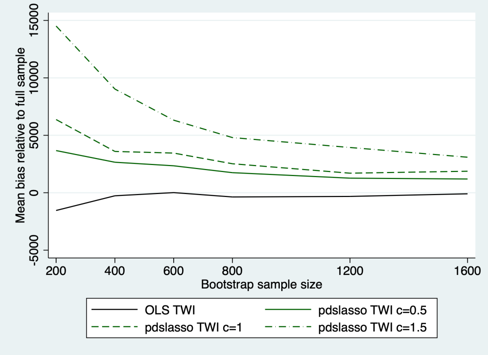
The bias in very small samples

A possible concern for machine learners is that they might not perform well for very small samples given that they are designed for, and typically applied to, large data sets.

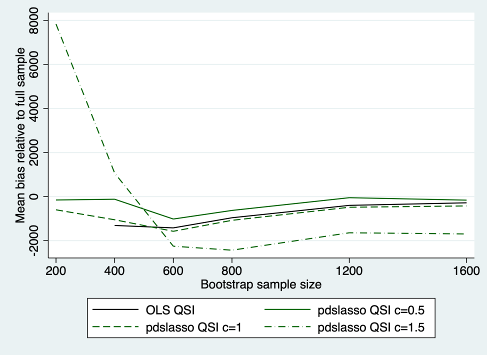
Wüthrich and Zhu (2021) use simulations to demonstrate that PDS-Lasso suffers from a significant small sample bias and tends to underselect.

Using the 401(k) data (Poterba et al., 1995), they consider two competing specifications: Two-way interactions (TWI) and Quadratic spline & interactions (QSI).

The bias in very small samples



(a) Bias (TWI)



(b) Bias (QSI)

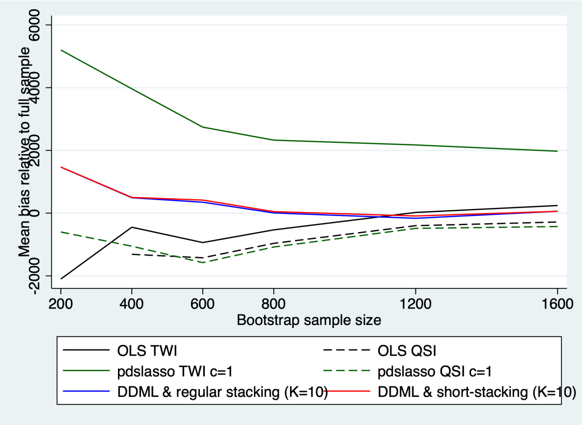
Notes: The figures report the mean bias calculated as the mean difference to the full sample estimates. Following WZ, we draw 600 bootstrap samples of size $n_b = \{200, 400, 600, 800, 1200, 1600\}$. 'TWI' indicates that the predictors have been expanded by two-way interactions. 'QSI' refers to the quadratic spline & interactions specification of Belloni et al. (2017).

Figure 3: Replication of Figure 8 in Wüthrich and Zhu (2021)

The bias in very small samples

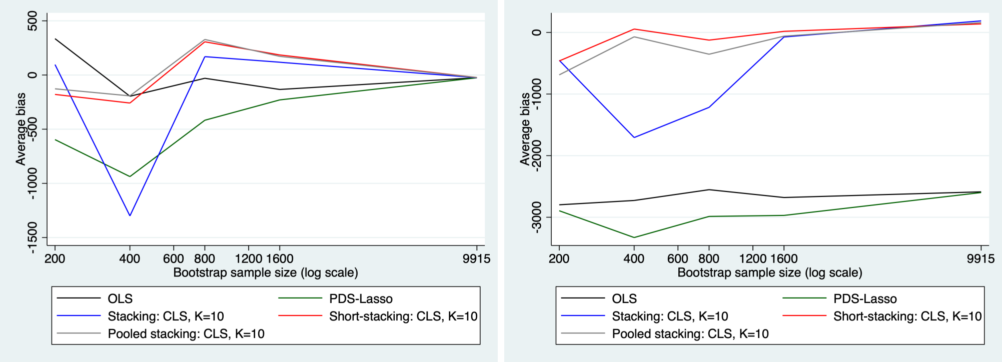
How do DDML paired with stacking or short-stacking perform in comparison?

Figure 4: Mean bias relative to full-sample estimates   Table version



Simulation II: The bias in very small samples

Figure 5: Bias in ***calibrated simulation*** Coverage Table



(a) Linear DGP (b) Non-linear DGP

DDML with stacking or short-stacking perform well even for moderate sample size. Increasing K improves performance especially for small samples.

Applications

We consider two applications that fall into the broader literature on unexplained gender gaps in various domains, e.g., entry to STEM programs (Card and Payne, 2021), ICT literacy (Siddiq and Scherer, 2019) or wages (Strittmatter and Wunsch, 2021; Bonaccolto-Töpfer and Briel, 2022).

- ▷ the gender wage gap in the UK
- ▷ the gender citation gap in International Relations ***(not today)*** Gender citation gap

Gender wage gap

- Country: UK
- Data: OECD
- Unconditional wage gap = -.1434 (s.e.=0.017)
- Number of observations = 4,889, $K = 10$
- AIPW estimator
- Covariates:
  - Categorical (21): part-time, industry, education, occupation, health status, management position, number of children, etc.
  - Continuous (5): age, tenure, literacy & numeracy, years of education
- Three sets of control specifications: "reduced" (only age, education, tenure), "baseline" (all variables) and "extended" (full interactions).

Gender wage gap

Details

Figure 7: Unexplained gender wage gap

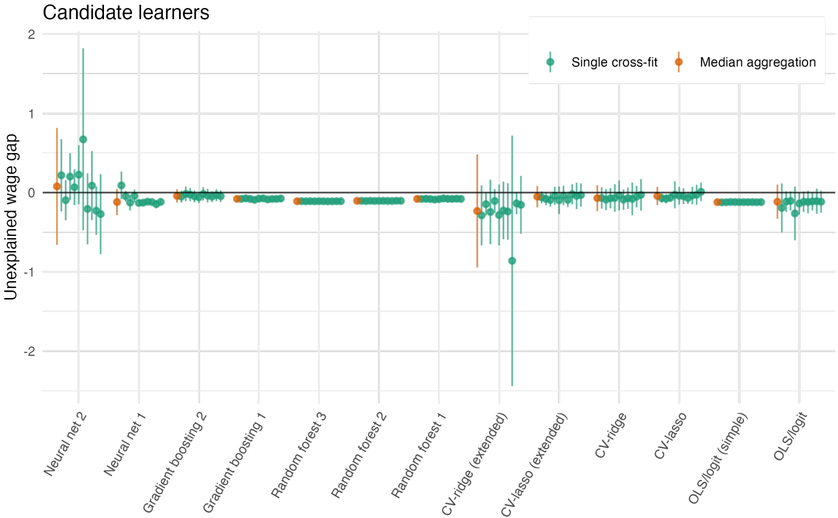


Figure 8: Unexplained gender wage gap

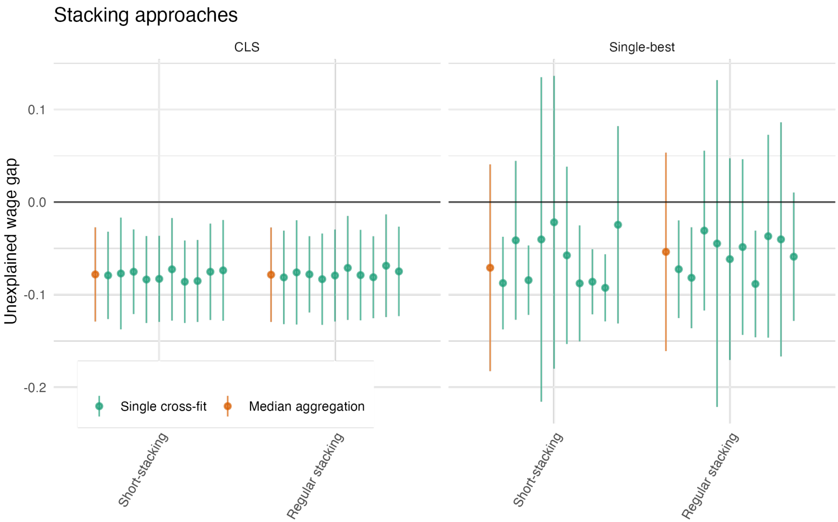


Table 4: Stacking weights in the gender wage gap application

	Stacking			*Short-stacking*		
	$g_0(0, X)$	$g_0(1, X)$	$m_0(X)$	$g_0(0, X)$	$g_0(1, X)$	$m_0(X)$
OLS/logit	0.027	0.010	0.246	0.037	0.009	0.218
OLS/logit (simple)	0.001	0.001	0.000	0.000	0.000	0.000
CV-lasso	0.111	0.136	0.109	0.060	0.073	0.063
CV-ridge	0.182	0.037	0.055	0.217	0.019	0.110
CV-lasso (extended)	0.035	0.189	0.005	0.007	0.253	0.015
CV-ridge (extended)	0.010	0.034	0.014	0.000	0.028	0.004
Random forest 1	0.440	0.499	0.289	0.479	0.498	0.284
Random forest 2	0.000	0.000	0.000	0.000	0.000	0.000
Random forest 3	0.000	0.000	0.000	0.000	0.000	0.000
Gradient boosting 1	0.026	0.009	0.025	0.028	0.008	0.017
Gradient boosting 2	0.145	0.050	0.227	0.155	0.081	0.269
Neural net 1	0.014	0.014	0.000	0.005	0.000	0.000
Neural net 2	0.009	0.021	0.030	0.012	0.031	0.021

Key recommendations

R1. Employ DDML paired with stacking or short-stacking with a diverse and generous set of candidate learners, including OLS.

R2. If the sample size is small, increase the number of folds and repeat the cross-fitting exercise.

R3. Inspect the (short-)stacking weights to adjust and refine learner settings.

Key takeaways

- DDML & stacking approaches ***safeguard against ill-chosen learners*** provided a diverse set of candidate learners is chosen.
- DDML paired with ***short-stacking performs comparably to regular stacking***—and in small samples even better.
- The ***lower computational cost*** is another advantage compared to DDML with regular stacking.
- Stacking weights are different for $E[Y|X]$ and $E[D|X]$, stressing the need to specify and tune machine learners separately for each CEF.

More info

Software

- Stata — The packages `ddml` and `pystacked` are available on Github/SSC. See `https://statalasso.github.io/` for info.
- R — The package `ddml` is available from CRAN. See `https://thomaswiemann.com/ddml/` for info.

References I

Ahrens, A., Hansen, C. B., and Schaffer, M. E. (2018). PDSLASSO: Stata module for post-selection and post-regularization OLS or IV estimation and inference. Medium: Statistical Software Components, Boston College Department of Economics.

Ahrens, A., Hansen, C. B., Schaffer, M. E., and Wiemann, T. (2023). ddml: Double/debiased machine learning in stata.

Angrist, J. D. and Frandsen, B. (2022). Machine labor. *Journal of Labor Economics*, 40(S1):S97–S140.

Belloni, A., Chernozhukov, V., Fernández-Val, I., and Hansen, C. (2017). Program Evaluation and Causal Inference With High-Dimensional Data. *Econometrica*, 85(1):233–298.

Belloni, A., Chernozhukov, V., and Hansen, C. (2014). Inference on treatment effects after selection among high-dimensional controls. *Review of Economic Studies*, 81:608–650.

Belloni, A., Chernozhukov, V., Hansen, C., and Kozbur, D. (2016). Inference in High Dimensional Panel Models with an Application to Gun Control. *Journal of Business & Economic Statistics*, 34(4):590–605. Genre: Methodology.

Bonaccolto-Töpfer, M. and Briel, S. (2022). The gender pay gap revisited: Does machine learning offer new insights? *Labour Economics*, 78:102223.

References II

Card, D. and Payne, A. A. (2021). High School Choices and the Gender Gap in Stem. *Economic Inquiry*, 59(1):9–28. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/ecin.12934.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68. tex.ids= Chernozhukov2018a publisher: John Wiley & Sons, Ltd (10.1111).

der Laan, M. J. V., Polley, E. C., and Hubbard, A. E. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1).

Giannone, D., Lenza, M., and Primiceri, G. E. (2021). Economic predictions with big data: The illusion of sparsity. *Econometrica*, 89(5):2409–2437.

Goller, D., Lechner, M., Moczall, A., and Wolff, J. (2020). Does the estimation of the propensity score by machine learning improve matching estimation? The case of Germany's programmes for long term unemployed. *Labour Economics*, 65:101855.

Grimmer, J., Roberts, M. E., and Stewart, B. M. (2022). *Text as data: A new framework for machine learning and the social sciences*. Princeton University Press.

References III

Maliniak, D., Powers, R., and Walter, B. F. (2013). The Gender Citation Gap in International Relations. *International Organization*, 67(4):889–922.

Polley, E. C., Rose, S., and van der Laan, M. J. (2011). Super learning. In *Targeted learning: Causal inference for observational and experimental data*, pages 43–66. Springer New York, New York, NY.

Poterba, J. M., Venti, S. F., and Wise, D. A. (1995). Do 401 (k) contributions crowd out other personal saving? *Journal of Public Economics*, 58(1):1–32.

Roberts, M. E., Stewart, B. M., and Nielsen, R. A. (2020). Adjusting for Confounding with Text Matching. *American Journal of Political Science*, 64(4):887–903.

Siddiq, F. and Scherer, R. (2019). Is there a gender gap? A meta-analysis of the gender differences in students' ICT literacy. *Educational Research Review*, 27:205–217.

Strittmatter, A. and Wunsch, C. (2021). The gender pay gap revisited with big data: Do methodological choices matter?

Van der Laan, M. J., Rose, S., et al. (2011). *Targeted learning: causal inference for observational and experimental data*, volume 4. Springer.

References IV

Wolpert, D. and Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390.

Wüthrich, K. and Zhu, Y. (2021). Omitted variable bias of lasso-based inference methods: A finite sample analysis. *Review of Economics and Statistics*, 0((0)):1–47.

Appendix: Coverage in very small samples

back

1. Construct the partial residuals $y_i^{(r)} = y_i - \hat{\tau}_{OLS} d_i$, $\forall i$ where $\hat{\tau}_{OLS}$ is the full sample OLS estiamte.
2. We predict $y_i^{(r)}$ with the controls x_i either using
   - linear regression (*Linear DGP*)
   - gradient boosting (*Non-Linear DGP*)

   and call the fitted estimator $\tilde{h}$.
3. Similarily, predict d_i given x_i and call the estimator $\tilde{g}$.
3. We draw bootstrap sample $\mathcal{D}_b$ of size n_s from the data
4. To generate 401(k) eligibility and log wealth, we calculate

$$\tilde{d}_i^{(b)} = \mathbb{1}\{\tilde{h}(x_i) + \nu_i \geq 0.5\}, \quad \nu_i \overset{iid}{\sim} \mathcal{N}(0, \kappa_1)$$
$$\tilde{y}_i^{(b)} = \tau_0 \tilde{d}_i^{(b)} + \tilde{g}(x_i) + \varepsilon_i, \quad \varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \kappa_2), \quad \forall i \in \mathcal{D}_b$$

   where κ_1 and κ_2 are chosen to match distributions of d_i and y_i.

Appendix: Advantages of DDML+Stacking

back

The table reports mean bias, median absolute bias (MAB) and coverage rate of a 95% confidence interval for the listed estimators. We consider DDML with the following candidate learners:

- ▷ OLS with elementary covariates,
- ▷ CV lasso and CV ridge with second-order polynomials and interactions,
- ▷ CV lasso and CV ridge with 10th-order polynomials but no interactions,
- ▷ random forest with low regularization: 8 predictors considered at each leaf split, no limit on the number of observations per node, bootstrap sample size of 70%,
- ▷ highly regularized random forest: 5 predictors considered at each leaf split, at least 10 observation per node, bootstrap sample size of 70%,
- ▷ gradient-boosted trees with low regularization: 500 trees and a learnings rate of 0.01,
- ▷ gradient-boosted trees with high regularization: 250 trees and a learnings rate of 0.01,
- ▷ feed-forward neural nets with three hidden layers of size five.

For reference, we report two estimators using the full sample: OLS and PDS lasso. We report results for four meta learners: Stacking with CLS, short-stacking with CLS, single best overall and single best by fold. Results are based on 1,000 replications.

Back

Table 5: Additional results: Bias and Coverage Rates in the ***Linear DGP***

	$n_b = 9{,}915$			$n_b = 99{,}150$		
Panel (A): Linear DGP	Bias	MAB	Rate	Bias	MAB	Rate
Full sample:						
OLS	−24.8	820.2	0.95	−1.9	269.1	0.95
PDS-Lasso	−25.4	821.2	0.95	0.6	269.4	0.95
DDML methods:						
Meta learners						
Stacking: CLS	8.8	829.3	0.95	−1.5	274.4	0.95
Stacking: Average	−0.1	812.9	0.94	1.4	273.7	0.95
Stacking: OLS	−33.7	874.1	0.94	2.6	272.0	0.95
Stacking: Single-best	−28.6	823.3	0.96	−3.5	272.1	0.95
Short-stacking: CLS	−23.8	817.6	0.96	−1.5	274.3	0.95
Short-stacking: Average	−0.1	812.9	0.94	1.4	273.7	0.95
Short-stacking: OLS	−27.9	818.1	0.96	−2.4	274.4	0.95
Short-stacking: Single-best	−20.7	826.0	0.96	−3.4	272.4	0.95
Pooled stacking: CLS	−24.8	817.8	0.96	−1.5	273.4	0.95
Pooled stacking: Average	−0.1	812.9	0.94	1.4	273.7	0.95
Pooled stacking: OLS	−48.1	834.6	0.95	−2.2	272.1	0.95
Pooled stacking: Single-best	−23.0	829.7	0.96	−3.3	270.9	0.95

Simulation I: Advantages of DDML+Stacking

Back

Table 6: Additional results: Bias and Coverage Rates in the ***Non-Linear DGP***

	n_b = 9,915			n_b = 99,150		
Panel (B): Non-Linear DGP	Bias	MAB	Rate	Bias	MAB	Rate
Full sample:						
OLS	−2587.3	2642.4	0.58	−2644.9	2640.5	0.
PDS-Lasso	−2598.4	2661.3	0.57	−2644.1	2638.4	0.
DDML methods:						
Meta learners						
Stacking: CLS	226.8	1129.3	0.93	24.9	262.9	0.95
Stacking: Average	−101.7	1102.7	0.93	60.0	271.9	0.95
Stacking: OLS	871.4	1264.7	0.93	26.6	265.7	0.94
Stacking: Single-best	179.2	1016.2	0.92	15.0	266.4	0.95
Short-stacking: CLS	221.9	897.3	0.93	21.9	261.7	0.95
Short-stacking: Average	−101.7	1102.7	0.93	60.0	271.9	0.95
Short-stacking: OLS	169.0	888.2	0.93	15.8	262.5	0.94
Short-stacking: Single-best	116.1	900.1	0.94	15.0	266.4	0.95
Pooled stacking: CLS	251.5	960.1	0.93	24.8	264.8	0.95
Pooled stacking: Average	−101.7	1102.7	0.93	60.0	271.9	0.95
Pooled stacking: OLS	376.8	1068.3	0.93	16.7	266.5	0.94
Pooled stacking: Single-best	131.0	952.0	0.93	15.0	266.4	0.95

Simulation I: Advantages of DDML+Stacking

back

Table 7: Computational time of DDML with regular and short-stacking

Folds K	Obs.	DDML Regular stacking	DDML Short-stacking	OLS	PDS-lasso	Ratio
2	200	24.69	6.34	0.0072	0.0617	0.2567
	400	26.02	6.76	0.0073	0.0640	0.2597
	800	29.51	7.67	0.0074	0.0665	0.2598
	1600	41.23	10.53	0.0082	0.0780	0.2554
	9915	210.78	53.01	0.0170	0.2131	0.2515
	99150	3434.07	778.17	0.1094	1.6571	0.2266
5	200	59.41	13.46	0.0069	0.0588	0.2266
	400	69.18	15.76	0.0070	0.0617	0.2278
	800	88.57	20.77	0.0074	0.0662	0.2345
	1600	137.77	31.92	0.0082	0.0781	0.2317
	9915	848.27	196.97	0.0148	0.1841	0.2322
10	200	120.47	26.01	0.0068	0.0583	0.2159
	400	141.29	30.95	0.0070	0.0608	0.2191
	800	189.87	42.98	0.0075	0.0677	0.2264
	1600	295.87	68.22	0.0082	0.0778	0.2306
	9915	1962.00	453.13	0.0159	0.1998	0.2310

Notes: The table reports the computational time in seconds of DDML paired with regular stacking or short-stacking as implemented in Ahrens et al. (2023), OLS as implemented in Stata's `regress`, post-double-selection lasso as implemented in `pdslasso` (Ahrens et al., 2018). DDML uses $V = 5$ cross-validation folds and K cross-fitting folds as indicated. Times reported are in seconds (average over 1,000 replications).

Table 8: Mean bias relative to full-sample estimates

	Bootstrap sample size n_b					
	200	400	600	800	1200	1600
Panel A. Full-sample estimators						
OLS QSI	−1680.9	−775	−806.4	−809.9	−677.2	−626.5
OLS TWI	−1542.3	−263.8	13.2	−366	−320.3	−91.3
Post double Lasso QSI c=0.5	444.7	−198.2	−204	−503.1	−571.6	−354.1
Post double Lasso QSI c=1	−149.8	−935.4	−639.4	−1063.2	−1000.5	−523.5
Post double Lasso QSI c=1.5	8153.6	724	−1526.2	−2434.4	−2255.4	−1863.5
Post double Lasso TWI c=0.5	3670.3	2656.4	2347.2	1748.3	1270.4	1197.5
Post double Lasso TWI c=1	6371.3	3594.6	3453.1	2523.9	1702.4	1871.8
Post double Lasso TWI c=1.5	14511.1	9026.1	6317.9	4802.2	3939	3094.5
Panel C. DDML with stacking approaches ($K = 10$)						
Stacking: CLS	1469.2	491.9	349.6	7.9	−163.4	62.1
Stacking: Single-best	787.7	140.3	113.9	−132	−309.1	52
Short-stacking: CLS	1470.3	501.3	417.7	48.6	−89.5	58.2
Short-stacking: Single-best	746.5	232.3	148.8	−178.1	−268.4	53.1

 back

Table 9: Mean bias in small samples based on the calibrated Monte Carlo

	Bootstrap sample size n_b									
	Panel A. Linear DGP					*Panel B. Non-linear DGP*				
	200	400	800	1600	9915	200	400	800	1600	9915
Full sample estimators:										
OLS	−246.2	−297.0	248.3	161.8	−20.2	−2291.4	−2023.2	−2631.2	−2857.2	−2582.1
PDS-Lasso	−1247.5	−1001.1	−159.3	72.0	−18.7	−2143.2	−2524.0	−3105.3	−3120.1	−2591.5
DDML methods:										
Base learners ($K = 10$)										
OLS	−282.8	−308.6	255.4	164.3	−19.9	−2833.5	−2364.7	−2861.6	−2952.6	−2600.8
Lasso with CV (2nd order poly)	−236.3	−206.1	277.8	174.3	−19.6	−912.8	−163.8	−82.8	264.4	754.2
Ridge with CV (2nd order poly)	−144.7	−187.2	303.5	151.2	−22.0	−1803.8	−728.0	122.1	697.7	778.0
Lasso with CV (10th order poly)	6339.9	180.0	1258.7	565.9	2.4	−7471.8	−3768.5	182.8	4054.5	−951.6
Ridge with CV (10th order poly)	5321.1	4716.5	6706.9	89.8	290.3	691.7	3137.6	−6506.9	−8867.9	515.8
Random forest (low regularization)	−149.1	−337.2	162.3	78.1	−110.7	−274.3	233.1	−448.8	−293.4	−15.0
Random forest (high regularization)	633.6	132.6	454.7	288.4	−15.3	−381.2	185.4	−361.3	−313.4	−74.3
Gradient boosting (low regularization)	−489.1	−498.4	125.6	113.6	−53.5	−539.0	182.4	−348.5	−209.5	59.7
Gradient boosting (high regularization)	−240.3	−284.2	291.9	257.5	40.8	−316.7	329.6	−159.4	−32.6	213.5
Neural net	3554.5	3955.7	3570.1	2153.1	129.7	1277.7	2089.6	1354.8	−132.9	−465.6
Meta learners ($K = 10$)										
Stacking: CLS	96.2	−1300.3	169.5	118.2	−24.9	−452.0	−1706.0	−1216.1	−73.8	188.5
Stacking: Single-best	−366.2	−1414.4	185.2	124.6	−25.2	−1774.8	−313.1	−622.0	−708.9	112.6
Short-stacking: CLS	−179.0	−258.7	306.6	185.3	−23.0	−462.7	55.0	−124.6	18.4	138.8
Short-stacking: Single-best	−308.2	−321.7	246.8	148.5	−22.6	−681.2	−149.9	−217.6	36.4	55.3

Appendix: Coverage in very small samples

back

Table 10: Coverage in small samples based on the calibrated Monte Carlo

	Bootstrap sample size n_b									
	Panel A. Linear DGP					*Panel B. Non-linear DGP*				
	200	400	800	1600	9915	200	400	800	1600	9915
Full sample estimators:										
OLS	0.95	0.95	0.96	0.96	0.95	0.94	0.95	0.92	0.91	0.59
PDS-Lasso	0.94	0.95	0.95	0.95	0.95	0.94	0.95	0.91	0.89	0.59
DDML methods:										
Base learners ($K = 10$)										
OLS	0.94	0.95	0.95	0.95	0.95	0.93	0.94	0.93	0.91	0.59
Lasso with CV (2nd order poly)	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.90
Ridge with CV (2nd order poly)	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.94	0.89
Lasso with CV (10th order poly)	0.90	0.92	0.93	0.93	0.95	0.87	0.85	0.85	0.88	0.95
Ridge with CV (10th order poly)	0.81	0.89	0.87	0.89	0.95	0.82	0.82	0.83	0.87	0.94
Random forest (low regularization)	0.92	0.92	0.93	0.92	0.91	0.93	0.92	0.93	0.93	0.91
Random forest (high regularization)	0.95	0.95	0.95	0.95	0.94	0.95	0.95	0.95	0.95	0.93
Gradient boosting (low regularization)	0.92	0.93	0.94	0.94	0.95	0.92	0.93	0.95	0.96	0.94
Gradient boosting (high regularization)	0.93	0.94	0.95	0.95	0.95	0.94	0.94	0.96	0.96	0.94
Neural net	0.94	0.92	0.90	0.91	0.95	0.94	0.95	0.95	0.97	0.94
Meta learners ($K = 10$)										
Stacking: CLS	0.93	0.94	0.95	0.95	0.95	0.94	0.93	0.94	0.94	0.94
Stacking: Single-best	0.93	0.94	0.95	0.95	0.95	0.94	0.95	0.95	0.94	0.94
Short-stacking: CLS	0.95	0.95	0.96	0.95	0.95	0.96	0.96	0.95	0.95	0.94
Short-stacking: Single-best	0.95	0.95	0.96	0.95	0.95	0.95	0.96	0.95	0.94	0.94

Appendix: Gender wage gap

back

Control variable sets:

- *base:* all continuous and categorical controls; interaction of age and tenure with categorical controls
- *reduced:* only region, industry and occupation as categorical controls; plus continuous controls
- *expanded:* full interactions of continuous and categorical controls

Candidate learner specifications:

- OLS/logit
- OLS/logit with reduced controls
- CV-lasso & CV-ridge
- CV-lasso & CV-ridge with expanded interactions
- RF with no restrictions of leaf size, 500 trees
- RF with minimum leaf size of 50, 500 trees
- RF with minimum leaf size of 100, 500 trees
- GB with early stopping after 10 rounds, maximum of 500 trees
- GB with 500 trees
- Neural net with hidden layer sizes of $(40, 20, 1, 20, 50)$ and early stopping
- Neural net with hidden layer sizes of $(30, 30, 30)$ and early stopping

Appendix: Gender citation gaps

back

Candidate learner specifications:

- CV-lasso
- CV-ridge
- XGBoost with maximum tree depth = 2, learning rate = 0.01, 1000 trees, early stopping
- XGBoost with maximum tree depth = 2, learning rate = 0.05, 1000 trees, early stopping
- XGBoost with maximum tree depth = 2, learning rate = 0.2, 1000 trees, early stopping
- XGBoost with maximum tree depth = 5, learning rate = 0.01, 1000 trees, early stopping
- XGBoost with maximum tree depth = 5, learning rate = 0.05, 1000 trees, early stopping
- XGBoost with maximum tree depth = 5, learning rate = 0.2, 1000 trees, early stopping
- Feedforward neural net with (10, 10, 10) units, dropout = 0.5, learning rate = .1
- Feedforward neural net with (10, 10, 10, 10) units, dropout = 0.5, learning rate = .1
- Feedforward neural net with (10, 10, 10, 10, 10) units, dropout = 0.5, learning rate = .1

Gender citation gap

back

We revisit *'The Gender Citation Gap in International Relations'* (Maliniak et al., 2013).

Outcome: citation count; treatment: all authors female. N=2,563 published articles.

Gender citation gap

back

We revisit *'The Gender Citation Gap in International Relations'* (Maliniak et al., 2013).

Outcome: citation count; treatment: all authors female. N=2,563 published articles.

Lower citations might reflect that all-female teams choose to work on different topics & methods, use different methods and/or language.

- ▷ Maliniak et al. (2013) control for tenure, hand-coded topic, journal, age of publication etc.
- ▷ Roberts et al. (2020) employ a text matching approach based on structural topic modeling.
- ▷ Grimmer et al. (2022) apply one-step CV-lasso with un-penalized treatment; applied to either words counts or topics.

Gender citation gap

back

We revisit *'The Gender Citation Gap in International Relations'* (Maliniak et al., 2013).

Outcome: citation count; treatment: all authors female. N=2,563 published articles.

Lower citations might reflect that all-female teams choose to work on different topics & methods, use different methods and/or language.

- Maliniak et al. (2013) control for tenure, hand-coded topic, journal, age of publication etc.
- Roberts et al. (2020) employ a text matching approach based on structural topic modeling.
- Grimmer et al. (2022) apply one-step CV-lasso with un-penalized treatment; applied to either words counts or topics.

We consider the partially linear model and two sets of controls: hand-coded controls (p=44), text as unigrams (90k).

Gender citation gap

back

Figure 9: The citation penalty for all-female authors

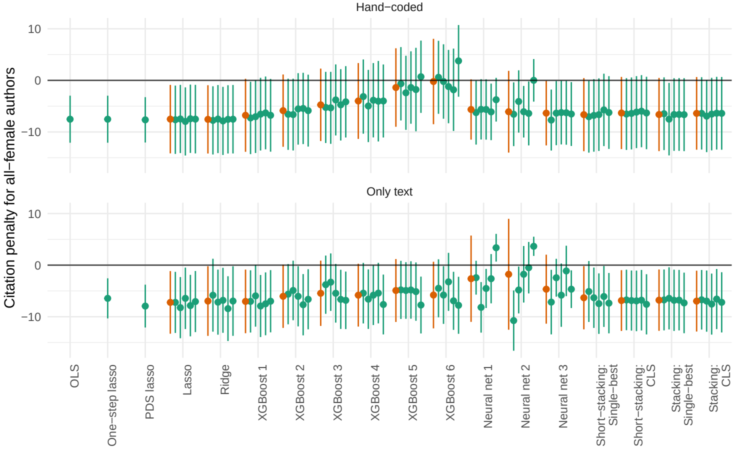


Table 11: Stacking weights in the gender citation gap application.

	Hand-coded				*Text only*			
	Regular		*Short-stacking*		*Regular*		*Short-stacking*	
	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$
Lasso	0.000	0.001	0.000	0.000	0.443	0.530	0.353	0.492
Ridge	0.407	0.000	0.333	0.000	0.005	0.099	0.000	0.009
XGBoost 1	0.179	0.891	0.088	0.901	0.000	0.124	0.000	0.232
XGBoost 2	0.226	0.000	0.422	0.000	0.009	0.064	0.003	0.017
XGBoost 3	0.081	0.002	0.043	0.000	0.198	0.079	0.148	0.135
XGBoost 4	0.024	0.103	0.081	0.097	0.001	0.022	0.000	0.000
XGBoost 5	0.006	0.001	0.004	0.000	0.220	0.022	0.350	0.019
XGBoost 6	0.019	0.002	0.000	0.000	0.119	0.038	0.146	0.062
Neural net 1	0.026	0.000	0.013	0.001	0.002	0.002	0.000	0.000
Neural net 2	0.021	0.000	0.000	0.001	0.001	0.009	0.000	0.032
Neural net 3	0.011	0.000	0.015	0.000	0.002	0.012	0.000	0.001

Notes: The table shows stacking weights for the considered candidate learners when using either hand-coded or text controls. The stacking weights are averaged over folds (in the case of regular stacking) and over cross-fitting repetitions.