Dear Members of the Search Committee,

You will find two of my recent papers in this PDF document:

1. "Model Averaging and Double Machine Learning" with Christian Hansen, Mark Schaffer, Thomas Wiemann, accepted in the *Journal of Applied Econometrics*

2. "Optimal multi-action treatment allocation: A two-phase field experiment to boost immigrant naturalization" with Alessandra Stampi-Bombelli, Selina Kurer, Dominik Hangartner, forthcoming in the *Journal of Applied Econometrics*

Thank you for your consideration.

Kind regards

Achim Ahrens

arXiv:2401.01645v2 [econ.EM] 25 Sep 2024

Model Averaging and Double Machine Learning*

Achim Ahrens† Christian B. Hansen‡ Mark E. Schaffer§

Thomas Wiemann‡

September 27, 2024

Abstract

This paper discusses pairing double/debiased machine learning (DDML) with *stacking*, a model averaging method for combining multiple candidate learners, to estimate structural parameters. In addition to conventional stacking, we consider two stacking variants available for DDML: *short-stacking* exploits the cross-fitting step of DDML to substantially reduce the computational burden and *pooled stacking* enforces common stacking weights over cross-fitting folds. Using calibrated simulation studies and two applications estimating gender gaps in citations and wages, we show that DDML with stacking is more robust to partially unknown functional forms than common alternative approaches based on single pre-selected learners. We provide Stata and R software implementing our proposals.

Keywords: causal inference, partially linear model, high-dimensional models, super learners, nonparametric estimation
JEL: C21, C26, C52, C55, J01, J08

Acknowledgment: Many thanks to Elliott Ash, Daniel Björkegren, David Cai, Ben Jann, Michael Knaus, Rafael Lalive, Moritz Marbach, Martin Huber, Blaise Melly, Gabriel Okasa, and Martin Spindler for helpful discussions and comments. We are also thankful for the helpful feedback we have received at the AI+Economics Workshop at the ETH Zürich in 2022, the Italian and Swiss Stata meetings in 2022, the 2022 Machine Learning in Economics Summer Institute in Chicago, the LISER workshop "Machine Learning in Program Evaluation, High-dimensionality and Visualization Techniques," the 2022 Scotland and Northern England Workshop in Applied Microeconomics, the IAAE Annual Conference in 2023, the London 2023 Stata meeting, the 2023 Stata Economics Virtual Symposium and the European Summer Meetings of the Econometric Society in 2023. We also thank anonymous reviewers for their feedback and suggestions. All remaining errors are our own. *Note:* An earlier version of the paper was presented under the title "A Practitioners' Guide to Double Machine Learning." *Conflict of interest:* The authors declare that they have no conflict of interest. *Data:* The authors provide replication code through the Journal of Applied Econometrics Data Archive and share data for all examples with the exception of the application in Section 5.1.

†Corresponding author. ETH Zürich, Leonhardshalde 21, 8092 Zürich, Switzerland. *Email:* achim.ahrens@gess.ethz.ch

‡University of Chicago, United States. *Email:* Christian.Hansen@chicagobooth.edu (Hansen), wiemann@uchicago.edu (Wiemann).

§Heriot-Watt University, Edinburgh, United Kingdom and IZA Institute of Labor Economics. *Email:* M.E.Schaffer@hw.ac.uk.

1 Introduction

Motivated by their robustness to partially unknown functional forms, supervised machine learning estimators are increasingly leveraged for causal inference. For example, lasso-based approaches such as the post-double-selection lasso (PDS lasso) of Belloni, Chernozhukov, and Hansen (2014) have become popular estimators of causal effects under conditional unconfoundedness in applied economics (e.g. Gilchrist and Sands, 2016; Dhar, Jain, and Jayachandran, 2022). Yet, a recent literature also raises practical concerns about the use of machine learning for causal inference. Wüthrich and Zhu (2023) find that lasso often fails to select relevant confounders in small samples while inference based on linear regression performs relatively well. Giannone, Lenza, and Primiceri (2021) and Kolesár, Müller, and Roelsgaard (2023) argue that the sparsity assumption, on which the lasso fundamentally relies, is frequently not plausible in economic data sets. Angrist and Frandsen (2022) show that conditioning on confounders using random forests may yield spurious results in IV regressions.[1] In an application to the evaluation of active labor market programs, Goller et al. (2020) find that random forests are not suitable for the estimation of propensity scores. A key characteristic shared by many of these studies using machine learning for causal inference is the focus on a single pre-selected machine learner.

This paper revisits the application of machine learning for causal inference in light of this recent literature. In particular, we highlight the benefits of pairing double/debiased machine learning (DDML) estimators of Chernozhukov et al. (2018) with stacking (Wolpert, 1996; Breiman, 1996; Laan, Polley, and Hubbard, 2007). DDML can leverage generic machine learners meeting mild convergence rate requirements for the estimation of common (causal) parameters. Stacking is a form of model averaging that allows selecting among or combining multiple candidate machine learners relying on different regularization assumptions rather than requiring an *ad hoc* choice between them. Based on a diverse set of applications and calibrated simulation studies, we show that the synthesis of stacking and DDML improves the robustness of estimates of target parameters to the underlying structure of the data, and illustrate the finite sample performance of stacking-based DDML estimators. The results suggest that stacking with a rich set of candidate estimators can address some of the shortcomings highlighted in the recent literature on causal inference

[1] See also Angrist (2022) for additional discussion.

with single pre-selected machine learners.

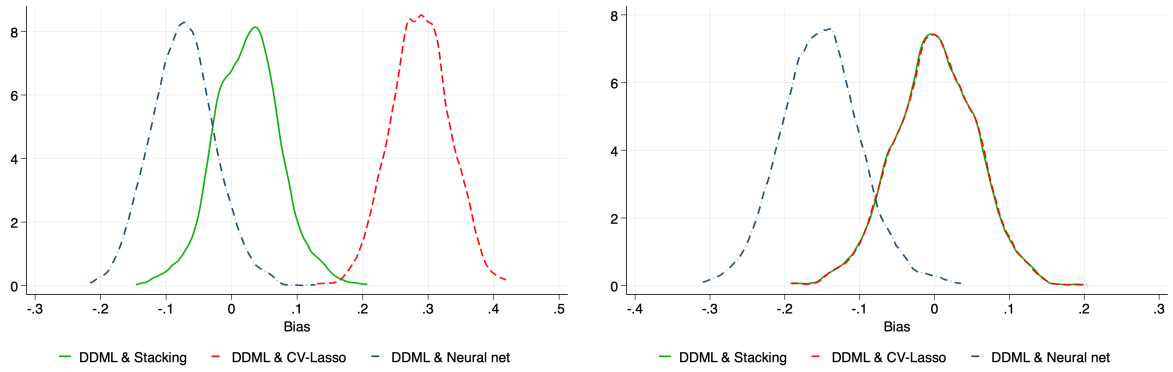
We further consider two alternate ways of combining stacking and DDML aimed at improving practical feasibility and stability in finite samples: *Short-stacking* leverages the cross-fitting step of DDML to reduce the computational burden of stacking substantially. *Pooled stacking* decreases the variance of stacking-based learners across the DDML cross-fitting folds. Both approaches facilitate interpretability compared to conventional stacking by enforcing common stacking weights. We complement the paper with software packages for Stata and R that implement the proposed approaches (Ahrens et al., 2024; Wiemann et al., 2024).

Model averaging techniques have a long tradition in economics and statistics. In the time-series literature, the idea of using 'optimal' weights to combine forecasts goes back to the 1960s (Crane and Crotty, 1967; Bates and Granger, 1969). Loss-minimizing combinations of a pre-specified set of estimators were introduced under the term stacking to the statistics literature by Wolpert (1992) and Breiman (1996) and generalized by Laan, Polley, and Hubbard (2007). Stacking fits a final (typically parametric) learner on a set of cross-validated predicted values derived from distinct candidate learners. A popular choice is to constrain the weights attached to each candidate learner to be non-negative and sum to one. The benefits of combining multiple estimators into a 'super learner' via stacking to improve robustness to the structure of the underlying data-generating process are well-known in the econometrics and statistics literature. Under appropriate restrictions on the data generating process and loss-function, Laan and Dudoit (2003) show asymptotic equivalence between stacking and the best-performing candidate learner.[2]

Model averaging methods and stacking are widely used in time-series forecasting and macro-econometrics (for recent reviews, see Steel, 2020; Wang et al., 2023). Yet, despite its theoretical appeal, stacking has hitherto been rarely used for the estimation of causal effects in economics or other social sciences. An important exception is Laan and Rose (2011), who recommend stacking in the context of Targeted Maximum Likelihood. Instead, estimators are often based on parametric (frequently linear) specifications or single pre-selected machine learners. This can have severe consequences for the properties of causal effect estimators if the given choice is ill-suited for the application at hand. A

[2]Hastie, Tibshirani, and Friedman (2009) and Laan and Rose (2011) provide textbook treatments of stacking and super learning. See also Hansen and Racine (2012) for discussion of jackknife (leave-one-out) stacking. Clydec and Iversen (2013) and Le and Clarke (2017) provide a Bayesian interpretation of stacking in the setting where the true model is not among the candidates.

Figure 1: Estimation bias of DDML with cross-validated lasso, feed-forward neural net and stacking



(a) Non-linear DGP (b) Approximately sparse linear DGP

Notes: The figures show kernel density plots comparing the bias of DDML paired with either cross-validated lasso, a feed-forward neural net (with two hidden layers of size 20) or a stacking learner combining 13 candidate learners (including cross-validated lasso and ridge, random forests, gradient-boosted trees and feed-forward neural nets). See Ahrens et al. (2024), where this example is taken from, for details on the specification of each learner. With respect to the data-generating processes, we generate 1000 samples of size $n = 1000$ using the PLM $Y_i = \theta_0 D_i + c_Y g(X_i) + \varepsilon_i$, $D_i = c_D g(X_i) + u_i$ where X_i are drawn from $\mathcal{N}(0, \Sigma)$ with $\Sigma_{i,k} = 0.5^{|j-k|}$. ε_i and u_i are drawn from standard normal distributions. In Figure (a), the nuisance function is $g(X_i) = X_{i,1}X_{i,2} + X_{i,3}^2 + X_{i,5}X_{i,5} + X_{i,6}X_{i,7} + X_{i,8}X_{i,9} + X_{i,10} + xX_{i,11}^2 + X_{i,12}X_{i,13}$. In Figure (b), the nuisance function is $g(X_i) = \sum_j 0.9^j X_{ij}$. c_Y and c_D are two constants chosen to ensure that the R^2 of the regression of Y onto X is approximately 0.5.

simple example is shown in Figure 1 which compares the performance of DDML using either cross-validated (CV) lasso or a feed-forward neural network to estimate a partially linear model across two different data-generating processes. The results show that the bias associated with each learner strongly depends on the structure of the data. Since true functional forms are often unknown in the social sciences, indiscriminate choices of machine learners in practice can thus result in poor estimates. DDML with stacking is a practical solution to this problem. As the example showcases, DDML using stacking is associated with low bias when considering a rich set of candidate learners that are individually most suitable to different structures of the data.

We conduct simulation studies calibrated to real economic datasets to demonstrate that stacking approaches can safeguard against ill-chosen or poorly tuned estimators in practical settings. Throughout, stacking estimators are associated with relatively low bias regardless of the simulated data-generating process, strongly contrasting the data-dependent performance of the causal effect estimators based on single pre-selected learners. The proposed stacking approaches thus appear relevant in the ubiquitous scenario where there is uncertainty about the set of control variables, correct functional form or the appropriate regularization assumption.

By revisiting the simulation design of Wüthrich and Zhu (2023), we further show that stacking can outperform linear regression for even small sample sizes. We argue that the poor small sample performance of lasso-based approaches is partially driven by the choice of covariate transformations and illustrate how stacking can accommodate a richer set of specifications, including competing parametric models. We also find that short-stacking and pooled stacking may outperform DDML paired with conventional stacking in small to moderate sample sizes. Paired with its lower computational cost, this finding suggests that short-stacking may be an attractive baseline approach to select and combine competing reduced form specifications.

Finally, we demonstrate the value of pairing of DDML with stacking with two applications. First, we examine gender gaps in citations of articles published in top-30 economic journals from 1983 to 2020, and assess how the difference in citations change when conditioning on content and quality proxied by the abstract text. Estimating these conditional differences is a challenging statistical problem due to the non-standard nature of text data, which is increasingly encountered in economic applications (see also e.g., Ash and Hansen, 2023; Ash, Chen, and Ornaghi, 2024; Eberhardt, Facchini, and Rueda, 2023). Second, we revisit a UK sample of the OECD Skill Survey to estimate semiparametric Kitagawa-Oxaca-Binder estimates of the unexplained gender wage gap. Both applications highlight that estimators of structural parameters based on single learners can be highly sensitive to the underlying structure of the data and/or poor tuning. The applications further demonstrate that DDML with stacking is a simple and practical solution to resolve the difficult problem of choosing a particular candidate learner in practice. Further, we observe that the optimal stacking weights often vary across reduced-form equations — meaning that different conditional expectation functions in the same data set are best estimated using different learners. This behavior sharply contrasts with common estimation approaches, such as OLS and PDS lasso, that impose the same form for each conditional expectation function.

The remainder of the paper is organized as follows: Section 2 provides a brief review of DDML. Section 3 discusses DDML with stacking, short-stacking, and pooled stacking. Section 4 presents our calibrated simulation studies. Section 5 discusses the applications, and Section 6 concludes.

2 Double/Debiased Machine Learning

This section outlines double/debiased machine learning as discussed in Chernozhukov et al. (2018). Throughout, we focus on the partially linear model as a natural extension of commonly applied linear regression methods. Despite its simplicity, the partially linear model illustrates practical challenges in the application of DDML that can be addressed by stacking. We highlight, however, that our discussion also applies to the wide range of models outlined in Chernozhukov et al. (2018) and more generally to estimation of low-dimensional structural parameters in the presence of high-dimensional nuisance functions.[3] Stacking could also be applied in settings where DDML is applied with nonparametric targets as in Colangelo and Lee (2023).

The partially linear model is defined by a random vector $(Y, D, X^\top, U)$ with joint distribution characterized by

$$Y = \theta_0 D + g_0(X) + U, \tag{1}$$

where Y is the outcome, D is the scalar variable of interest, and X is a vector of control variables. The parameter of interest θ_0 and the unknown nuisance function g_0 are such that the corresponding residual U satisfies the conditional orthogonality property $E[Cov(U, D|X)] = 0$. These properties are analogous to the orthogonality properties of residuals in multiple linear regression with the key difference here being that g_0 need not be linear in the controls.

Albeit a seemingly small change in specification, the partially linear model has several important advantages over linear regression. For discrete D, for example, results in Angrist and Krueger (1999) imply that θ_0 can be interpreted as a positively weighted average of incremental changes in the conditional expectation function $E[Y|D = d, X]$. Under appropriate conditional unconfoundedness assumptions, θ_0 thus corresponds to a convex combination of conditional average treatment effects.[4] Importantly, these inter-

[3] A key example not explicitly discussed in Chernozhukov et al. (2018) is doubly-robust estimation of difference-in-difference parameters with staggered treatment assignment as in Callaway and Sant'Anna (2021) and Chang (2020). In settings with conditional parallel trends assumptions, high-dimensional nuisance functions arise in the estimation of group-time specific average treatment effect on the treated. The pairing of DDML and stacking, as proposed in this paper, also directly applies to the estimator of Callaway and Sant'Anna (2021) under a conditional unconfoundedness assumption.

[4] Similarly, for continuous D, θ_0 corresponds to a positively weighted average of derivatives of the conditional expectation function $E[Y|D = d, X]$ with respect to d. Under a conditional unconfoundedness assumption, θ_0 is thus a convex combination of derivatives of the causal response function.

pretations remain valid even if the additive separability assumption of the partially linear model fails. Linear regression coefficients, in contrast, do not correspond to positively weighted averages of causal effects without imposing strong linearity assumptions that are questionable in real applications.[5]

The advantages of the partially linear model in the interpretation of its parameter of interest come at the cost of a more challenging estimation problem relative to estimating a model that is linear in a pre-specified set of variables. Estimators for θ_0 are based on the solution to the moment equation

$$E\left[\left(Y - \ell_0(X) - \theta_0(D - m_0(X))\right)\left(D - m_0(X)\right)\right] = 0,$$

given by

$$\theta_0 = \frac{E\left[\Big(Y - \ell_0(X)\Big)\Big(D - m_0(X)\Big)\right]}{E\left[(D - m_0(X))^2\right]},$$

where $\ell_0(X) \equiv E[Y|X]$ and $m_0(X) \equiv E[D|X]$ are the conditional expectations of the outcome and variable of interest given the controls, respectively. Since conditional expectation functions are high-dimensional in the absence of strong functional form assumptions, a sample analogue estimator for θ_0 requires nonparametric first-step estimators for the nuisance parameters ℓ_0 and m_0. While nonparametric estimation generally reduces bias compared to linear regression alternatives, the increased variance associated with more flexible functional form estimation introduces additional statistical challenges: To allow for statistical inference on θ_0, the nonparametric estimators need to converge sufficiently quickly to the true conditional expectation functions as the sample size increases.

DDML defines a class of estimators that allows for statistical inference on the parameter of interest θ_0 while only imposing relatively mild convergence requirements on the nonparametric estimators. These mild requirements are central to the wide applicability of DDML as they permit the use of a large variety of machine learners.[6]

[5] In the context of IV estimation where instrument validity relies on observed confounders, Blandhol et al. (2022) emphasize that, in the absence of strong functional form assumptions, two stage least squares does not generally correspond to a convex combination of local average treatment effects (LATE). We note that the IV analogue to the partially linear model does admit a causal interpretation under the LATE assumptions, just as the partially linear model admits a weakly causal interpretation under conditional unconfoundedness.

[6] The exact convergence rate requirement for nonparametric estimators depends on the parameter of interest. Chernozhukov et al. (2018) name the crude rate requirement of $o(n^{-1/4})$, but provide examples

Two key devices permit the mild convergence requirements of DDML: Identification of the parameter of interest based on Neyman-orthogonal moment conditions and estimation using cross-fitting. Neyman-orthogonal moment conditions are insensitive to local perturbations around the true nuisance parameter.[7] Cross-fitting is a sample-splitting approach that addresses the *own-observation bias* that arises when the nuisance parameter estimation and the estimation of θ_0 are applied to the same observation. In practice, cross-fitting is implemented by randomly splitting a sample $\{(Y_i, D_i, X_i^\top)\}_{i \in I}$ indexed by $I = \{1, \ldots, n\}$ into K evenly-sized folds, denoted as $I_1, \ldots, I_K$. For each fold k, the conditional expectations ℓ_0 and m_0 are estimated using only observations not in the kth fold — i.e., in $I_k^c \equiv I \setminus I_k$ — resulting in $\hat{\ell}_{I_k^c}$ and $\hat{m}_{I_k^c}$, respectively, where the subscript I_k^c indicates the subsample used for estimation. The out-of-sample predictions for an observation i in the kth fold are then computed via $\hat{\ell}_{I_k^c}(X_i)$ and $\hat{m}_{I_k^c}(X_i)$. Repeating this procedure for all K folds then allows for computation of the DDML estimator for θ_0:

$$\hat{\theta}_n = \frac{\frac{1}{n}\sum_{i=1}^n \left(Y_i - \hat{\ell}_{I^c_{k_i}}(X_i)\right)\left(D_i - \hat{m}_{I^c_{k_i}}(X_i)\right)}{\frac{1}{n}\sum_{i=i}^n \left(D_i - \hat{m}_{I^c_{k_i}}(X_i)\right)^2},$$

where k_i denotes the fold of the ith observation.

Since the cross-fitting algorithm depends on the randomized fold split, and since some machine learners rely on randomization too, DDML estimates vary with the underlying random-number generator and seed. To reduce dependence on randomization, it is thus worthwhile to repeat the cross-fitting procedure and apply mean or median aggregation over DDML estimates (see Remark 2 in Ahrens et al., 2024). We show in Section 5 that repeating the cross-fitting procedure is a useful diagnostic tool, allowing to gauge the stability of DDML estimators.

---

where the rate requirement is considerably weaker. Recent contributions show that these requirements are satisfied by specific instances of machine learners; see, e.g., results for lasso (Bickel, Ritov, and Tsybakov, 2009; Belloni et al., 2012), random forests (Wager and Walther, 2016; Wager and Athey, 2018; Athey, Tibshirani, and Wager, 2019), neural networks (Schmidt-Hieber, 2020; Farrell, Liang, and Misra, 2021), and boosting (Luo, Spindler, and Kück, 2022). The exact asymptotic properties of many other machine learners remain an active research area.

[7] In the context of the partially linear model, the formal Neyman-orthogonality requirement is

$$\begin{aligned} 0 = \frac{\partial}{\partial \lambda} E\Big[&\big(Y - \{\ell_0(X) + \lambda(\ell(X) - \ell_0(X))\} - \tau_0(D - \{m_0(X) + \lambda(m(X) - m_0(X))\})\big) \\ &\times \big(D - \{m_0(X) + \lambda(m(X) - m_0(X))\}\big)\Big]\Big|_{\lambda=0} \end{aligned}$$

for arbitrary measurable functions ℓ and m, which can easily be verified using properties of the residuals.

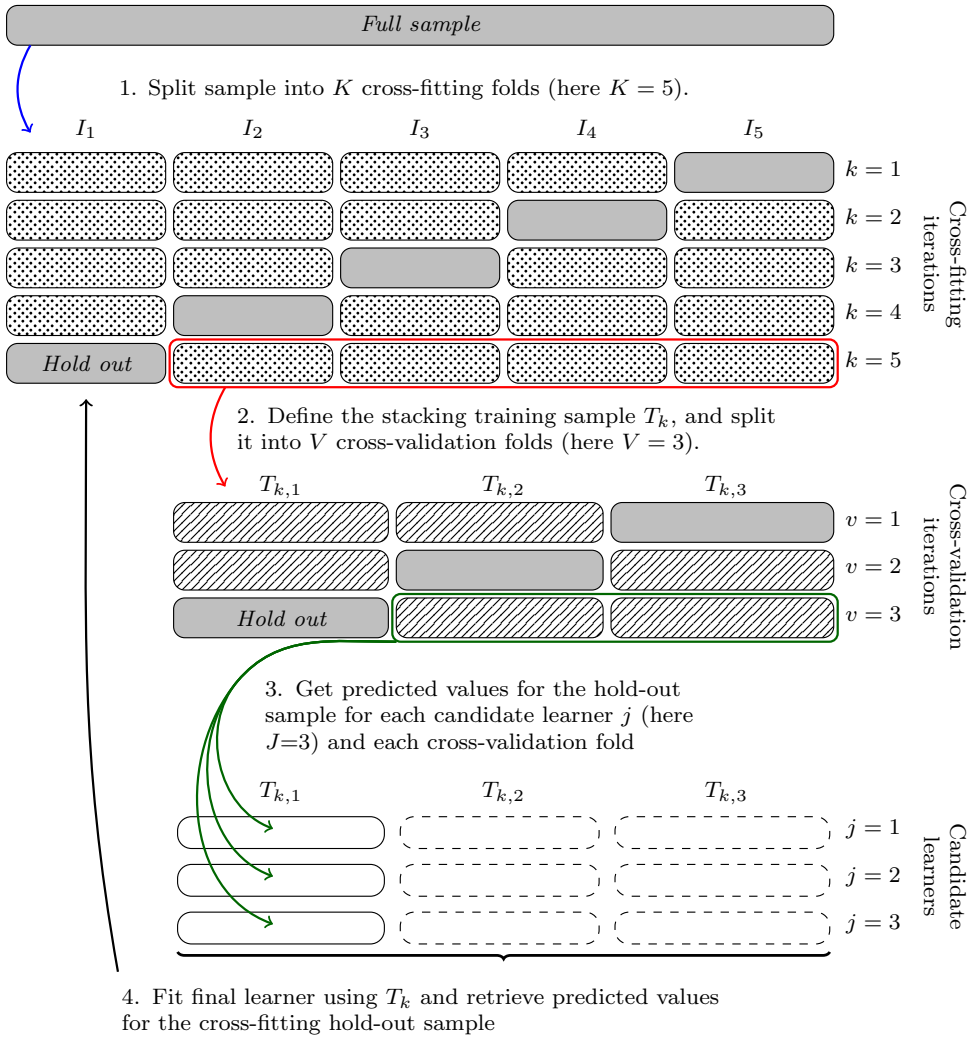
Under the conditions of Chernozhukov et al. (2018) — including, in particular, the convergence requirements on the nonparametric estimators — $\hat{\theta}_n$ is root-n asymptotically normal around θ_0. As already highlighted by the example in Figure 1, however, a poorly chosen or poorly tuned machine learner for the estimation of nuisance parameters $\hat{\ell}$ and $\hat{m}$ can have detrimental effects on the properties of $\hat{\theta}_n$. Since no machine learner can be best across all settings, this raises the difficult question of which learner to apply in a particular setting. In the next section, we discuss how DDML can be paired with stacking to provide a practical solution to the choice of learner. We also illustrate how the cross-fitting structure naturally arising in DDML estimators can be leveraged to substantially reduce the computational burden otherwise associated with stacking.

3 Pairing DDML with Stacking Approaches

This section discusses the estimation of structural parameters by pairing DDML with stacking approaches. After the discussion of DDML with conventional stacking, we introduce two stacking variants that leverage the cross-fitting structure of DDML estimators: short-stacking and pooled stacking. To fix ideas, we focus on the nuisance parameter $\ell_0(X) = E[Y|X]$ arising in the partially linear model where we consider an i.i.d. sample $\{(Y_i, X_i)\}_{i \in I}$. Further, we consider a rich set of J pre-selected base or candidate learners. The set of learners could include distinct parametric and nonparametric estimators — e.g., linear or logistic regression, regularized regression such as the lasso, or tree-based methods such as random forests — as well as the same algorithm with varying (hyper-) tuning parameters or different (basis) expansions of the control variables. It is important to note that the set of candidate learners for stacking can readily incorporate commonly used unregularized learners such as linear or logistic regression; in practice, sometimes the best-performing candidate learner may be one such learner.

DDML with conventional stacking. Combining DDML with conventional stacking involves two layers of re-sampling, as we illustrate in Figure 2. The *cross-fitting layer* divides the sample into K cross-fitting folds, denoted by $I_1, \ldots, I_K$. In each cross-fitting step $k \in \{1, \ldots, K\}$, the stacking learner is trained on the training sample which excludes fold I_k and which we label $T_k \equiv I \setminus I_k$. Fitting the stacking learner, in turn, requires subdividing the training sample T_k further into V cross-validation folds. This second sample

Figure 2: Cross-fitting with conventional stacking



Notes: The diagram illustrates cross-fitting with conventional stacking. The diagram uses $K = 5$ cross-fitting folds, $V = 3$ cross-validation folds and $J = 3$ candidate learners. Step 1: The sample randomly is split into cross-fitting folds $I_1, \ldots, I_K$. Step 2: In each step $k \in \{1, \ldots, K\}$ of the cross-fitting process, we define the training data as $T_k \equiv I \setminus I_k$. The k-step training data is then split into V sub-partitions, which we denote by $T_{k,1}, \ldots, T_{k,V}$. Step 3: For each cross-fit step k and cross-validation step $v \in \{1, \ldots, V\}$, fit each base learner $j \in \{1, \ldots, J\}$ on $T_k \setminus T_{k,v}$ and obtain predicted values for the cross-validation hold-out sample. Step 4: Fit the final learner on sample T_k to obtain predicted values for the cross-fitting hold-out sample I_k.

split constitutes the *cross-validation layer*. We denote the cross-validation folds in cross-fitting step k by $T_{k,1}, \ldots, T_{k,V}$. Each candidate learner $j \in \{1, \ldots, J\}$ is cross-validated on these folds, yielding cross-validated predicted values for each learner.

The final learner fits the outcome Y_i against the cross-validated predicted values of each candidate learner. The most common choice is to construct a convex combination via constrained least squares (CLS), with weights restricted to be non-negative and summing to one. Specifically, for each k, candidate learners are combined to solve

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

Here, $\hat{\ell}^{(j)}_{T^c_{k,v(i)}}(X_i)$ denotes the out-of-sample predicted value for observation i, which is calculated from training candidate learner j on sub-sample $T^c_{k,v(i)} \equiv T_k \setminus T_{k,v(i)}$, i.e., all step-$k$ cross-validation folds but fold $(k, v(i))$ which is the fold of the ith observation. We call the resulting $\hat{w}_{k,j}$ the *stacking weights*. The stacking predictions are obtained as $\sum_j \hat{w}_{k,j} \hat{\ell}^{(j)}_{T_k}(X_i)$ where each learner j is re-fit on T_k.

Although various options for combining candidate learners are available, CLS facilitates the interpretation of stacking as a weighted average of candidate learners (Hastie, Tibshirani, and Friedman, 2009). Due to this constraint, CLS tends to set some stacking weights to exactly zero. The constraint also regularizes the final estimator, which is important to mitigate issues arising from potential multicollinearity of the candidate learners. An alternative to CLS, which we refer to as *single-best learner*, is to impose the constraint that $w_{k,j} \in \{0, 1\}$ and $\sum_j w_{k,j} = 1$, implying that only the candidate learner with lowest cross-validated loss is used as the final estimator. Under appropriate restrictions on the data-generating process and loss function, Laan and Dudoit (2003) show asymptotic equivalence between stacking and the best-performing candidate learner.[8]

A drawback of DDML with stacking is its computational complexity. Considering the estimation of a single candidate learner as the unit of complexity (and ignoring the cost of fitting the final learner), DDML with stacking heuristically has a computational cost proportional to $K \times V \times J$. For example, when considering DDML with $K = 5$ cross-fitting folds and $J = 10$ candidate learners that are combined based on $V = 5$ fold cross-

[8]The *scikit-learn* (Buitinck et al., 2013) routines `StackingRegressor` and `StackingClassifier` implement stacking for Python. In Stata, stacking regression and classification are available via `pystacked`, which is a Stata front-end for these Python routines (Ahrens, Hansen, and Schaffer, 2023).

validation, more than 250 candidate learners need to be individually estimated. Although DDML with stacking is "embarrassingly parallel" and can thus be expected to decrease in computational time nearly linearly in the number of available computing processes, the increased complexity limits its application to moderately complex applications. Another potential concern (which we investigate in Section 4.2) is that DDML with stacking might not perform well in small samples, given that candidate learners are effectively trained on approximately $\frac{(K-1)(V-1)}{KV}\%$ of the full sample (see Figure 2). These two concerns motivate *short-stacking*.

DDML with short-stacking. In the context of DDML, we propose to take a shortcut: Instead of fitting the final learner on the cross-*validated* fitted values in each step k of the cross-fitting process, we can directly train the final learner on the cross-*fitted* values using the full sample; see Figure 3. Formally, candidate learners are combined to solve

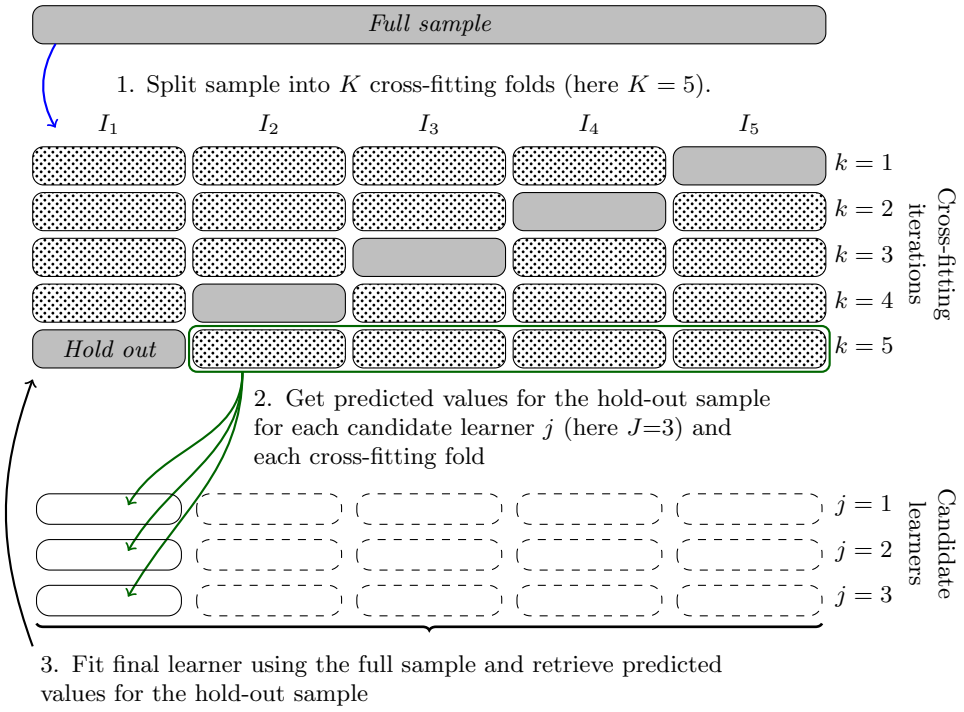
$$\min_{w_1,\ldots,w_J} \sum_{i=1}^{n} \left(Y_i - \sum_{j=1}^{J} w_j \hat{\ell}^{(j)}_{I^c_{k(i)}}(X_i) \right)^2 \qquad \text{s.t. } w_j \geq 0,\ \sum_j |w_j| = 1$$

where w_j are the short-stacking weights. Cross-fitting thus serves a double purpose: First, it avoids the own-observation bias by avoiding overlap between the samples used for estimating high-dimensional nuisance functions and the samples used for estimating structural parameters. Second, it yields out-of-sample predicted values which we leverage for constructing the final stacking learner. As a consequence, the computational cost of DDML with short stacking is heuristically only proportional to $K \times J$ in units of estimated candidate learners. In the example from the previous paragraph, short-stacking thus requires estimating about 200 fewer candidate learners.

We recommend DDML with short-stacking in settings where the number of candidate learners is small relative to the sample size, i.e., $J \ll n$. We believe this setting provides a good approximation to current applications of machine learning in economics and other social sciences where it is rare to consider more than a few candidate learners. If instead the number of considered learners is very large relative to the sample size — i.e., settings in which inference for standard linear regression on J variables is invalid — pairing DDML with short-stacking may introduce bias.[9]

[9]Suppose, for simplicity, we consider ordinary (unconstrained) least squares as the final learner. Heuristically, the regression of Y_i against J sets of cross-fitted predicted values is akin to a conventional least

Figure 3: Cross-fitting with *short*-stacking



Notes: The diagram illustrates cross-fitting with short-stacking. The diagram uses $K = 5$ cross-fitting folds and $J = 3$ candidate learners. Step 1: The sample randomly is split into cross-fitting folds $I_1, \ldots, I_K$. Step 2: In each step $k \in \{1, \ldots, K\}$ of the cross-fitting process, we define the training data as $I_k^c = I \setminus I_k$. Fit each base learner $j \in \{1, \ldots, J\}$ on the training data and obtain predicted values for the cross-fitting hold-out sample I_k. Step 3: Fit the final learner on the full sample to obtain predicted values for the cross-fitting hold-out sample.

DDML with pooled stacking. While DDML with conventional stacking has one vector of weights per cross-fitting fold, short-stacking yields a single weight for each learner. A single weight for each learner decreases the variance of the final estimator and facilitates the interpretation of the stacking weights. Another way of achieving common stacking weights is DDML with pooled stacking. Pooled stacking relies on the same two-layer re-sampling strategy as conventional stacking, but combines candidate learners to solve

$$\min_{w_1,\ldots,w_J} \sum_{i\in I} \sum_{k\neq k(i)} \left(Y_i - \sum_{j=1}^{J} w_j \hat{\ell}^{(j)}_{T^c_{k,v(i)}}(X_i) \right)^2 \qquad \text{s.t. } w_j \geq 0,\ \sum_{j=1}^{J} |w_j| = 1.$$

That is, pooled stacking collects the cross-validated predicted values that are calculated in each step k of the cross-fitting process for each learner j and estimates the stacking weights based on the pooled data set. We note that the computational costs are approximately the same as for DDML with conventional stacking.

squares regression of Y_i against J observed regressors where good performance would require $J/n \to 0$, ignoring that the cross-fitted predicted values are estimated. The additional regularization by constrained least squares should further weaken this rate requirement.

4 The Practical Benefits of DDML with Stacking: Two Simulation Studies

In this section, we discuss two simulation studies illustrating the advantages of pairing DDML with stacking over alternative approaches based on single pre-selected learners. We begin with a simulation calibrated to household data on wealth and 401k eligibility from the 1991 wave of the Survey of Income and Program Participation (SIPP) in Section 4.1. In Section 4.2, we revisit the simulation of Wüthrich and Zhu (2023) to assess the robustness of DDML with stacking approaches in very small samples.

4.1 Simulation calibrated to the SIPP 1991 household data

To assess the performance of DDML with conventional stacking, short-stacking and pooled stacking in a realistic setting, we consider the analysis of 401(k) eligibility and total financial assets in Poterba, Venti, and Wise (1995) as the basis for an empirically calibrated Monte Carlo simulation. The application has recently been revisited by Belloni et al. (2017), Chernozhukov et al. (2018), and Wüthrich and Zhu (2023) to approximate high-dimensional confounding factors using machine learning. We focus on estimating the partially linear model discussed in the previous section. The outcome is measured as net financial assets, the treatment variable is an indicator for eligibility to the 401(k) pension scheme, and the set of controls includes age, income, education in years, family size, as well as indicators for two-earner status, home ownership, and participation in two alternative pension schemes.

The simulation involves three steps. In the calibration step, we fit two generative models to the $n = 9\,915$ households from the 1991 wave of the Survey of Income and Program Participation. The first generative model is fully linear while the second is partially linear, allowing controls to enter non-linearly through gradient-boosted trees fitted to the real data. This approach is aimed at extracting and magnifying the linear or non-linear structures in the empirical conditional distributions, respectively, enabling us to compare the performance of estimators across favorable and unfavorable structures of the data. The generative step then simulates datasets of size $n_b = \{9\,915,\ 99\,150\}$ from the respective fully linear model and the partially linear model. Throughout, we set the effect of 401(k) eligibility on total financial wealth to $\theta_0 = 6\,000$. Finally, in the estimation step,

Let $\{(y_i, d_i, x_i)\}_{i=1,\dots,n}$ denote the observed sample, where i is a household in the 1991 SIPP and y_i, d_i, and x_i respectively denote net financial assets, an indicator for 401(k) eligibility, and the vector of control variables.

1. Using the full sample, obtain the slope coefficient $\hat{\theta}_{OLS} \approx 5\,896$ from linear regression of d_i against d_i, and x_i in the original data. Construct the partial residuals $y_i^{(r)} = y_i - \hat{\theta}_{OLS} d_i$, $\forall i$.
2. Fit a supervised learning estimator (either linear regression or gradient boosting) to predict $y_i^{(r)}$ with the controls x_i. Denote the fitted estimator by $\tilde{g}$. Similarly, fit a supervised learning estimator to predict d_i with x_i and denote the fitted estimator by $\tilde{h}$.
3. Repeat to generate simulated samples of size n_b:
   (a) Sample from the empirical distribution of x_i by bootstrapping n_b observations from the original data. Denote the bootstrapped sample by $\mathcal{D}_b$.
   (b) Draw $\nu_i \overset{iid}{\sim} \mathcal{N}(0, \kappa_1)$ and $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \kappa_2)$, where κ_1 and κ_2 are simulation hyperparameters. Define

$$\tilde{d}_i^{(b)} = \mathbb{1}\{\tilde{h}(x_i) + \nu_i \geq 0.5\}$$
$$\tilde{y}_i^{(b)} = \theta_0 \tilde{d}_i^{(b)} + \tilde{g}(x_i) + \varepsilon_i \qquad \forall i \in \mathcal{D}_b$$

   where we set $\theta_0 = 6\,000$ to roughly resemble the magnitude of the regression coefficient of 401(k) eligibility in the full data.

Notes: We set the hyper-parameter κ_1 and κ_2 to approximately match variance of 401(k) eligibility and wealth in the data. The values of the simulation hyperparameters (κ_1, κ_2) differ slightly depending on the supervised learning estimator used to fit the reduced form equations in the data. We take $\kappa_1 = 0.35$ in both scenarios but take $\kappa_2 = 55\,500$ when using linear regression and $\kappa_2 = 54\,000$ when using gradient boosting. Differences arise because gradient boosting reduces residual variance in the true data.

Algorithm 1: Algorithm for the calibrated Monte Carlo simulation

we fit various estimators to bootstrapped samples of the generated datasets and assess their statistical properties. We outline the steps used for constructing the two generative models in more detail in Algorithm 1.

For each bootstrap sample, we calculate estimates of the effect of 401(k) eligibility on simulated net financial assets. The estimators we consider are linear regression, the post-double selection (PDS) lasso estimator proposed by Belloni, Chernozhukov, and Hansen (2014), as well as DDML estimators with and without stacking. The candidate learners of the DDML estimators are linear regression, cross-validated lasso and ridge regression with interactions and second-order polynomial expansions of the controls, cross-validated lasso and ridge with no interactions but 10^{th}-order polynomial expansions of the controls, two versions of random forests, two versions of gradient-boosted trees, and feed-forward neural nets with three hidden layers of size five (see Table 1 notes for details). We estimate DDML paired with conventional stacking, short-stacking and pooled stacking,

and consider different methods to construct the final conditional expectation function estimator: CLS, unconstrained linear regression (OLS), selecting the single best estimator, and an unweighted average.

Table 1 presents the mean bias, median absolute bias (MAB) and coverage rates of a 95% confidence interval associated with estimates of the effect of 401(k) eligibility on net financial assets. The left and right panels correspond to results based on data simulated from the linear (Panel A) and non-linear (Panel B) generative models, respectively. The CLS weights associated with each candidate learner are shown in Table 2.[10]

Given the construction of the generative models, we would expect that linear regression performs best in the fully linear setting and that DDML with gradient boosting performs best in the nonlinear setting where the nuisance function is generated by gradient boosting. The simulation results confirm this intuition, showing that the two procedures achieve among the lowest bias and median absolute bias in the data-generating processes that are based on them. Researchers are rarely certain of the functional structure in economic applications, however, so that it is more interesting to consider their respective performance in the non-favorable setting. In the non-linear data-generating process, linear regression is among the estimators with the worst performance across all three measures. Similarly, gradient boosting-based DDML is non-optimal in the linear data-generating process. It is outperformed by linear regression and CV lasso, both of which enforce a linear functional form on the control variables, in terms of MAB.

The simulation results are consequences of the "no free lunch" theorem in machine learning (Wolpert, 1996). Informally, the theorem states that there exists no estimator that performs best across all empirical settings. Researchers must, therefore, carefully match estimators to their application. However, with limited knowledge about underlying data-generating processes and few functional form restrictions implied by economic theory, the number of plausibly suitable estimators is typically large.

The bottom section of Table 1 reports results for DDML combined with the three stacking approaches outlined in Section 3. For each stacking approach, we consider stacking weights estimated by (CLS) as outlined in Section 3, set equal to $1/J$ (Average), estimated without constraint by OLS (OLS), and by selecting only the single best candidate learner

[10]Further results are provided in the Appendix. Table A.1 in the Appendix gives the mean-squared prediction errors (MSPE) for each candidate learner for comparison. Table A.2 reports standard errors of the bias. Tables A.3 and A.4 show the stacking weights when using single-best and OLS as the final learner, respectively.

Table 1: Bias and Coverage Rates in the Linear and Non-Linear DGP

	Panel (A): Linear DGP						*Panel (B): Non-linear DGP*					
	$n_b = 9\,915$			$n_b = 99\,150$			$n_b = 9\,915$			$n_b = 99\,150$		
	Bias	MAB	Rate	Bias	MAB	Rate	Bias	MAB	Rate	Bias	MAB	Rate
Full sample:												
OLS	49.9	793.8	0.95	−6.8	281.2	0.95	−2588.9	2576.5	0.58	−2632.3	2611.5	0.
PDS-Lasso	48.4	787.1	0.95	−4.2	280.8	0.95	−2598.7	2590.1	0.58	−2631.6	2609.5	0.
DDML methods:												
Candidate learners												
OLS	46.2	818.1	0.94	−6.9	283.1	0.95	−2613.0	2634.2	0.58	−2635.4	2615.9	0.
Lasso with CV (2nd order poly)	50.9	806.6	0.95	−6.2	284.8	0.95	703.7	1052.3	0.91	718.5	712.8	0.60
Ridge with CV (2nd order poly)	48.2	806.9	0.94	−6.9	283.7	0.96	767.4	1080.8	0.90	729.3	724.0	0.60
Lasso with CV (10th order poly)	248.1	1034.5	0.94	55.9	285.9	0.95	−4109.0	1799.9	0.90	7.4	306.5	0.94
Ridge with CV (10th order poly)	1230.1	1321.9	0.91	31.6	283.0	0.96	−5126.2	2215.7	0.89	9.6	307.8	0.94
Random forest (low regularization)	−74.7	1031.3	0.89	−25.2	344.0	0.88	−96.1	1037.1	0.90	−37.5	328.0	0.87
Random forest (high regularization)	69.1	891.2	0.94	−23.5	287.6	0.93	−159.7	904.4	0.94	−4.2	280.4	0.95
Gradient boosting (low regularization)	12.1	817.0	0.94	−24.2	285.1	0.96	8.5	866.0	0.94	30.9	275.1	0.96
Gradient boosting (high regularization)	114.8	823.8	0.94	66.9	285.6	0.95	162.0	857.2	0.94	200.1	314.6	0.93
Neural net	394.2	943.6	0.93	9.1	287.5	0.94	−601.3	1063.9	0.93	−131.9	310.0	0.93
Stacking approaches												
Stacking: CLS	42.8	813.4	0.94	−7.5	282.9	0.96	133.9	1049.5	0.94	37.8	271.0	0.95
Stacking: Average	107.7	821.9	0.94	−6.5	273.6	0.95	94.0	1035.6	0.94	72.3	289.3	0.96
Stacking: OLS	−129.3	878.7	0.94	−9.4	283.8	0.95	−204.8	1184.9	0.94	17.5	268.3	0.96
Stacking: Single-best	43.7	819.8	0.94	−8.6	281.4	0.95	−121.9	976.2	0.94	30.9	275.1	0.96
Short-stacking: CLS	45.0	794.9	0.94	−7.0	282.6	0.95	162.7	865.1	0.94	33.6	266.3	0.95
Short-stacking: Average	107.7	821.9	0.94	−6.5	273.6	0.95	94.0	1035.6	0.94	72.3	289.3	0.96
Short-stacking: OLS	37.6	803.7	0.94	−7.8	282.0	0.95	123.6	863.5	0.94	29.5	265.3	0.96
Short-stacking: Single-best	44.4	817.8	0.94	−8.3	281.9	0.95	71.7	868.4	0.94	30.9	275.1	0.96
Pooled stacking: CLS	58.6	819.5	0.95	−7.1	283.9	0.96	209.8	921.0	0.94	37.5	269.8	0.95
Pooled stacking: Average	107.7	821.9	0.94	−6.5	273.6	0.95	94.0	1035.6	0.94	72.3	289.3	0.96
Pooled stacking: OLS	46.5	805.2	0.94	−7.8	284.0	0.96	234.5	1003.2	0.94	30.6	266.3	0.96
Pooled stacking: Single-best	46.9	823.5	0.94	−8.3	282.6	0.95	103.3	904.6	0.94	30.9	275.1	0.96

Notes: 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 $K = 2$ cross-fit folds and the following individual 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, maximum depth of 3 and a learning rate of 0.01), gradient-boosted trees with high regularization (250 trees, maximum depth of 3 and a learning 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. Finally, we report results for DDML paired with conventional stacking, short-stacking and pooled stacking where the final estimator is either CLS, OLS, the unweighted average of candidate learners or the single-best candidate learner. Results are based on 1 000 replications.

(Single-best). We find that short-stacking performs similarly to, and sometimes better than, conventional and pooled stacking, while being computationally much cheaper (as shown in Table A.5). For example, at $K = 10$ and $V = 5$, DDML combined with short-stacking ran around 4.3 times faster on the full sample than DDML with conventional or pooled stacking, which is roughly in line with a speed improvement by a factor of $1/V$.[11]

The simulation set-up is favorable to using single-best as the final learner because there is one 'true' candidate learner. However, single-best does not visibly outperform CLS, although single-best always selects the correct learner in the non-linear DGP if the sample size is sufficiently large (see Appendix Table A.4). We believe that CLS is, in practice, a good default choice. In a setting where there is a single learner that does a distinctly better job approximating the target conditional expectation function, CLS should assign a very large weight to that learner and thus approximate single-best. Otherwise, when there are several learners with similar performance or learners that perform differentially well for different observations, there are potential gains from combining the different learners.

The bias of the OLS final learner is overall similar to CLS, except when employing conventional stacking under the non-linear DGP for $n_b = 9\,915$ where the average bias is almost four times as large.[12] The unweighted average appears sub-optimal for $n_b = 99\,150$ under the non-linear DGP. Poor performance of unweighted averaging is to be expected in settings where, as in our setup, the candidate set includes learners that are not well-matched to the DGP.[13] We also note that the computational advantage of short-stacking with the unweighted average over short-stacking with estimated weights amounts to one (constrained) regression per conditional expectation function and is thus minimal.

The CLS weights in Table 2 indicate that stacking approaches successfully assign the highest weights to the estimators aligning with the data-generating process (i.e., either OLS or gradient boosting) among the ten included candidate learners, illustrating the

[11] The computations were performed on the high-performance cluster of the ETH Zurich. Each instance used a single core of an AMD EPYC processor with 2.25-2.6GHz (nominal)/3.3-3.5 GHz (peak) and 4GB RAM. The run time of DDML with conventional stacking was 2 393s on the full sample, while short-stacking ran in only 540s.

[12] Appendix Table A.3 shows that the OLS stacking weights are often outside the unit interval. The weights associated with the neural net are particularly large (in absolute value), suggesting that OLS might be more sensitive to outliers than CLS.

[13] By contrast, the unweighted average is known to often perform well in time-series settings (e.g., Clemen, 1989; Timmermann, 2006), and in particular when the optimal weights are close to equal (see Wang et al. (2023), Section 2.6 for a summary and wider discussion). Such a scenario is ruled out in our simulation setup, which captures the situation where the researcher does not know whether a linear or non-linear learner would be more appropriate.

Table 2: Average stacking weights with CLS

	Stacking		Pooled stacking		Short-stacking	
	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$
Panel (A): Linear DGP and $n_b = 9,915$						
OLS	0.668	0.501	0.738	0.573	0.692	0.492
Lasso with CV (2nd order poly)	0.105	0.144	0.093	0.131	0.118	0.130
Ridge with CV (2nd order poly)	0.068	0.054	0.050	0.040	0.068	0.063
Lasso with CV (10th order poly)	0.027	0.073	0.021	0.059	0.020	0.085
Ridge with CV (10th order poly)	0.033	0.043	0.018	0.027	0.024	0.057
Random forest (low regularization)	0.013	0.011	0.009	0.007	0.009	0.008
Random forest (high regularization)	0.017	0.024	0.012	0.018	0.013	0.024
Gradient boosting (low regularization)	0.030	0.043	0.024	0.034	0.020	0.040
Gradient boosting (high regularization)	0.020	0.060	0.020	0.070	0.018	0.060
Neural net	0.019	0.049	0.014	0.041	0.018	0.043
Panel (B): Linear DGP and $n_b = 99,150$						
OLS	0.770	0.325	0.824	0.348	0.769	0.291
Lasso with CV (2nd order poly)	0.066	0.026	0.055	0.012	0.068	0.013
Ridge with CV (2nd order poly)	0.074	0.041	0.057	0.028	0.094	0.049
Lasso with CV (10th order poly)	0.015	0.207	0.011	0.225	0.011	0.198
Ridge with CV (10th order poly)	0.023	0.111	0.018	0.095	0.019	0.129
Random forest (low regularization)	0.003	0.003	0.002	0.002	0.002	0.002
Random forest (high regularization)	0.006	0.009	0.005	0.006	0.005	0.006
Gradient boosting (low regularization)	0.019	0.160	0.015	0.175	0.013	0.186
Gradient boosting (high regularization)	0.004	0.008	0.003	0.004	0.003	0.003
Neural net	0.019	0.110	0.009	0.106	0.016	0.122
Panel (C): Non-linear DGP and $n_b = 9,915$						
OLS	0.011	0.015	0.007	0.012	0.004	0.007
Lasso with CV (2nd order poly)	0.035	0.057	0.024	0.051	0.019	0.039
Ridge with CV (2nd order poly)	0.161	0.229	0.192	0.268	0.114	0.237
Lasso with CV (10th order poly)	0.053	0.080	0.047	0.071	0.048	0.062
Ridge with CV (10th order poly)	0.071	0.064	0.060	0.029	0.059	0.056
Random forest (low regularization)	0.045	0.011	0.043	0.006	0.043	0.005
Random forest (high regularization)	0.019	0.069	0.010	0.065	0.012	0.065
Gradient boosting (low regularization)	0.521	0.233	0.548	0.251	0.632	0.339
Gradient boosting (high regularization)	0.014	0.191	0.005	0.203	0.004	0.139
Neural net	0.071	0.051	0.065	0.043	0.064	0.049
Panel (D): Non-linear DGP and $n_b = 99,150$						
OLS	0.	0.	0.	0.	0.	0.
Lasso with CV (2nd order poly)	0.	0.	0.	0.	0.	0.
Ridge with CV (2nd order poly)	0.	0.036	0.	0.037	0.	0.026
Lasso with CV (10th order poly)	0.	0.001	0.	0.	0.	0.
Ridge with CV (10th order poly)	0.	0.036	0.	0.034	0.	0.024
Random forest (low regularization)	0.153	0.003	0.154	0.001	0.180	0.001
Random forest (high regularization)	0.	0.060	0.	0.063	0.	0.071
Gradient boosting (low regularization)	0.845	0.853	0.846	0.858	0.819	0.871
Gradient boosting (high regularization)	0.	0.	0.	0.	0.	0.
Neural net	0.001	0.012	0.	0.006	0.001	0.007

Notes: The table shows the average stacking weights associated with the candidate learner for DDML with conventional stacking, pooled stacking and short-stacking. The final learner is CLS. The bootstrap sample size is denoted by n_b. The number of cross-fitting folds is $K = 2$. Results are based on 1 000 replications. See Table 1 for more information. The final learner weights using OLS and single best are reported in Appendix Tables A.3 and A.4.

ability to adapt to different data structures. Specifically, the stacking methods applied to the linear data-generating process assign the largest weight to linear models while they assign the largest weights to the gradient-boosting estimators and the lowest weights to estimators that impose a linear functional form on the control variables in the non-linear data-generating process.[14] We conclude that DDML paired with stacking approaches reduces the burden of choice researchers face when selecting between candidate learners and specifications by allowing for the simultaneous consideration of multiple options, thus implying attractive robustness properties across a variety of data-generating processes.

4.2 DDML and Stacking in Very Small Samples

A possible concern for estimators relying on machine learning is that they might not perform well for very small samples, given that their flexibility comes at the cost of increased variance compared to parametric estimators. Wüthrich and Zhu (2023, henceforth WZ) use two simulations to demonstrate that PDS lasso tends to underselect controls, which may result in a substantial small-sample bias. They also show that the bias heavily depends on the exact lasso penalty chosen (i.e., whether the plugin penalty of Belloni, Chernozhukov, and Hansen, 2014, is scaled by 0.5 or 1.5), and argue in favor of OLS with appropriately chosen standard errors over PDS lasso in high-dimensional settings.

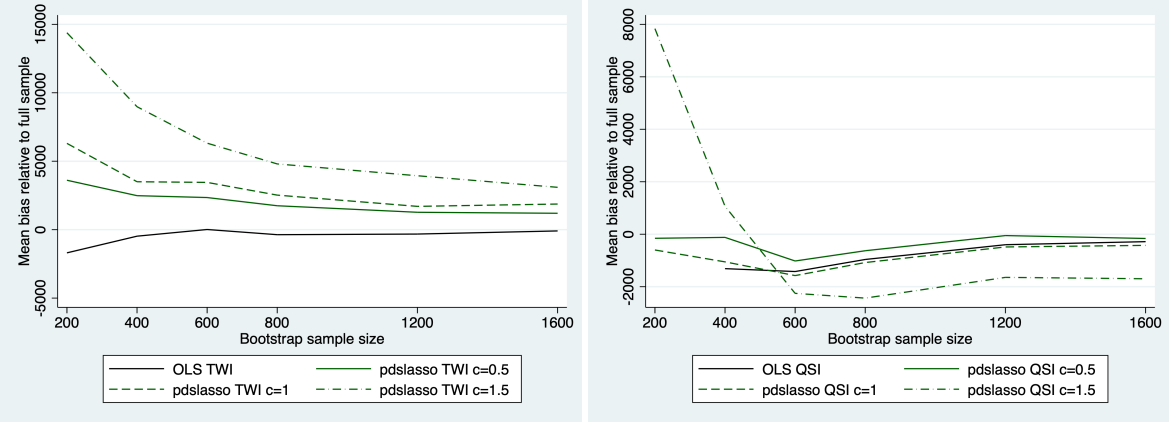
We revisit the 401(k) simulation set-up in WZ to assess if DDML with stacking suffers from similar issues in small samples and to compare the performance of DDML paired with stacking with PDS lasso and OLS. Following WZ, we run simulations on bootstrap samples of the data for $n_b = \{200, 400, 800, 1\,600\}$ and approximate the bias as the mean difference relative to the full-sample estimates ($n = 9\,915$).[15] WZ consider two sets of controls: two-way interactions (TWI), and quadratic splines with interactions (QSI) (as in Belloni et al., 2017). The number of predictors is 167 and 272, respectively. Figure 4 replicates the main results of WZ (Figure 8 in their paper). Panels (a) and (b) show the bias relative to the full sample estimate for the TWI and QSI specification based on OLS and PDS lasso with tuning parameter equal to the plugin penalty of Belloni, Chernozhukov, and Hansen (2014) scaled by c for $c \in \{0.5, 1, 1.5\}$. It is noteworthy that the speed at which the bootstrapped estimates converge to the full-sample estimate

[14] The rates at which each candidate learner is selected by the single-best final learner are shown in Table A.4 in the appendix and provide similar insights.

[15] The full-sample estimates are reported in Table B.1.

depends on the set of controls for the PDS lasso, but less so for OLS. While PDS lasso with $c = \{0.5, 1\}$ and OLS perform similarly if QSI controls are used, PDS lasso converges much more slowly to the full-sample estimate with TWI controls.

Figure 4: Replication of Figure 8 in Wüthrich and Zhu (2023).



(a) Bias (TWI) (b) Bias (QSI)

Notes: The figures report the mean bias calculated as the mean difference to the full sample estimates. Full sample estimates reported in Table B.1. Following WZ, we draw 1 000 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).

The DDML-stacking framework allows us to choose between, and combine, OLS and lasso with both the TWI and QSI set of controls. Another advantage of DDML over PDS lasso is that we can leverage lasso with cross-validated penalization for a fully data-driven penalization approach. Table 3 compares the performance of the full-sample estimators OLS and PDS lasso (shown in Panel A) to DDML-stacking estimators only relying on OLS and CV lasso with TWI and QSI controls as candidate learners (Panel B). We again consider conventional stacking, short-stacking and pooled stacking together with either CLS or single-best as the final learner. We set the number of cross-fitting folds to $K = 10$ (but also consider $K = 2$ below for comparison in Panel E).

Across all sample sizes, the DDML-stacking estimators strictly outperform both OLS specifications, as well as PDS lasso with TWI, and exhibit overall similar performance to PDS lasso utilizing QSI controls and $c = \{0.5, 1\}$. The differences across DDML-stacking estimators are relatively minor. The CLS short-stacking weights reported in Table 4, Panel A-B, reveal that CV-lasso with QSI controls receives the largest weights, while both OLS specifications contribute jointly between nearly zero (at $n_b = 200$) and only up to 15% (for $n_b = 1\,600$ and the estimation of $E[D|X]$). When selecting only a single candidate learner, CV-lasso with QSI is chosen in more than three-fourths of

Table 3: 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	−2083.5	−910.2	−806.4	−809.9	−677.2	−626.5
OLS TWI	−1694.5	−475.4	13.2	−366	−320.3	−91.3
Post double Lasso QSI c=0.5	409.2	−308.9	−204	−503.1	−571.6	−354.1
Post double Lasso QSI c=1	−179.1	−1113.5	−639.4	−1063.2	−1000.5	−523.5
Post double Lasso QSI c=1.5	8021.3	739.9	−1526.2	−2434.4	−2255.4	−1863.5
Post double Lasso TWI c=0.5	3611.2	2484.4	2347.2	1748.3	1270.4	1197.5
Post double Lasso TWI c=1	6303.3	3501.1	3453.1	2523.9	1702.4	1871.8
Post double Lasso TWI c=1.5	14 386.1	8981.9	6317.9	4802.2	3939	3094.5
Panel B. DDML-stacking with only OLS and CV lasso ($K = 10$)						
Short-stacking: CLS	1020	−113.8	−181.1	−538.2	−575.6	−292.4
Short-stacking: Single-best	1002.3	−122.2	−270.1	−499.7	−550.3	−197.7
Pooled stacking: CLS	925.7	−237.3	−319.1	−628.1	−711	−370.5
Pooled stacking: Single-best	782.3	−200.5	−358.9	−541.2	−580.2	−237.5
Stacking: CLS	1155.8	−254.7	−266.9	−645	−633	−315.1
Stacking: Single-best	999.5	−23.6	−184.9	−503.9	−571.1	−248.2
Panel C. DDML-stacking will all candidate learners ($K = 10$)						
Short-stacking: CLS	1355.1	342.2	403.3	34.2	−103.9	43.8
Short-stacking: Single-best	669.2	113.5	144.6	−182.3	−272.6	48.9
Pooled stacking: CLS	2849.3	1345.7	1197	383.8	−102.3	−10.6
Pooled stacking: Single-best	724.1	−69.4	45	−250.7	−309	−19.4
Stacking: CLS	1394.1	296.9	344.5	2.8	−168.5	56.9
Stacking: Single-best	718.4	−47	104.3	−141.5	−318.6	42.5
Panel D. DDML with candidate learners ($K = 10$)						
OLS	963	−150.8	210	−161.7	−235.5	31.8
Lasso with CV (TWI)	5948.6	3223.1	2589.1	1706.2	872.2	734.1
Ridge with CV (TWI)	4137.3	1853.8	1617.5	951.8	657.5	879.2
Lasso with CV (QSI)	297.5	−343.9	−311.9	−551.8	−597.1	−239.8
Ridge with CV (QSI)	426.1	−111	85.3	−240.8	−294.4	−7.8
Random forest (low regularization)	1852.8	618.3	709.6	259.7	7.7	95.5
Random forest (high regularization)	9987.4	4270.1	2940.2	1919.5	1037.8	925
Gradient boosting (low regularization)	772.3	−25	306.3	70.7	−127.2	113
Gradient boosting (high regularization)	1060.8	94.3	564.6	292.5	44.2	228.6
Neural net	8892.3	7481.2	6915.4	5653.2	3716.5	2224.2
Panel E. DDML-stacking will all candidate learners ($K = 2$)						
Short-stacking: CLS	1842.3	1078.3	−144.4	61.2	446.7	282.9
Short-stacking: Single-best	1303.5	582.3	−436.4	−248.8	194	111.1
Pooled stacking: CLS	2799	1471.3	159.5	209.8	572.7	508.8
Pooled stacking: Single-best	1791.9	622.9	−542.3	−296.3	144.7	84.8
Stacking: CLS	1924.6	1196.1	−191.2	59.4	390.3	310.9
Stacking: Single-best	1173.4	549.6	−604.2	−285	181.8	138.3

Notes: The table reports the mean bias calculated as the mean difference to the full sample estimates. Following WZ, we draw 1 000 bootstrap samples of size n_b. In Panel A, we show results for the full-sample estimators OLS and PDS lasso using either two-way interactions as controls (denoted TWI) or the quadratic spline & interactions specification of Belloni et al. (2017, denoted as QSI). We scale the PDS lasso penalty by $c = 0.5$, 1 or 1.5. In Panel B, we report results for DDML with stacking approaches and only relying on OLS and CV lasso. In Panel C, we consider a larger set of candidate learners. These are: OLS, CV lasso and CV ridge with either TWI or QSI controls, 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%) or high regularization (5 splitting predictors, at least 10 observation per node, bootstrap sample size of 70%), gradient-boosted tree with either low (500 trees, learning rate of 0.01, maximum depth of 3) or high (250 trees, learning rate of 0.01, maximum depth of 3) regularization, and a neural net with three hidden layers of size 5. Panel D shows results for these individual candidate learners. In Panels B–D, we use $K = 10$ cross-fitting folds and $R = 5$ cross-fitting repetitions. Panel D uses the same specifications as Panel C, but uses $K = 2$.

bootstrap iterations for the estimation of $E[Y|X]$ and $E[D|X]$ (Panel C-D in Table 4), suggesting that CV-lasso with QSI controls is strictly preferable over OLS and lasso with TWI controls in this application. This simulation exercise again highlights that relying on poorly chosen specifications that are not validated against other choices might be sub-optimal. In practice, the researcher does not know whether TWI or QSI controls perform better and whether to use OLS or lasso. Crucially, DDML paired with stacking allows for simultaneous consideration of OLS and lasso with both TWI and QSI controls and thus resolves the choice between learners and control specifications in a data-driven manner.

Table 4: Short-stacking weights

Estimator	*Observations*						
	200	400	600	800	1 200	1 600	9 915
Panel A. Constrained least squares. $E[Y\|X]$, $K = 10$							
OLS (TWI)	.01	.042	.062	.078	.098	.113	.013
OLS (QSI)	0	0	.002	.008	.023	.032	.128
Lasso with CV (TWI)	.249	.2	.196	.171	.158	.14	.214
Lasso with CV (QSI)	.74	.758	.74	.742	.721	.716	.645
Panel B. Constrained least squares. $E[D\|X]$, $K = 10$							
OLS (TWI)	.005	.037	.055	.074	.1	.127	.13
OLS (QSI)	0	0	.001	.003	.011	.022	.134
Lasso with CV (TWI)	.264	.163	.137	.119	.114	.111	.232
Lasso with CV (QSI)	.731	.8	.807	.803	.775	.74	.504
Panel C. Single-best. $E[Y\|X]$, $K = 10$							
OLS (TWI)	0	0	0	0	.001	0	0
OLS (QSI)	0	0	0	0	0	0	0
Lasso with CV (TWI)	.186	.141	.128	.112	.09	.081	0
Lasso with CV (QSI)	.814	.859	.872	.888	.909	.919	1
Panel D. Single-best. $E[D\|X]$, $K = 10$							
OLS (TWI)	0	0	0	0	0	0	0
OLS (QSI)	0	0	0	0	0	0	0
Lasso with CV (TWI)	.239	.126	.098	.079	.06	.068	.003
Lasso with CV (QSI)	.761	.874	.902	.921	.94	.932	.997

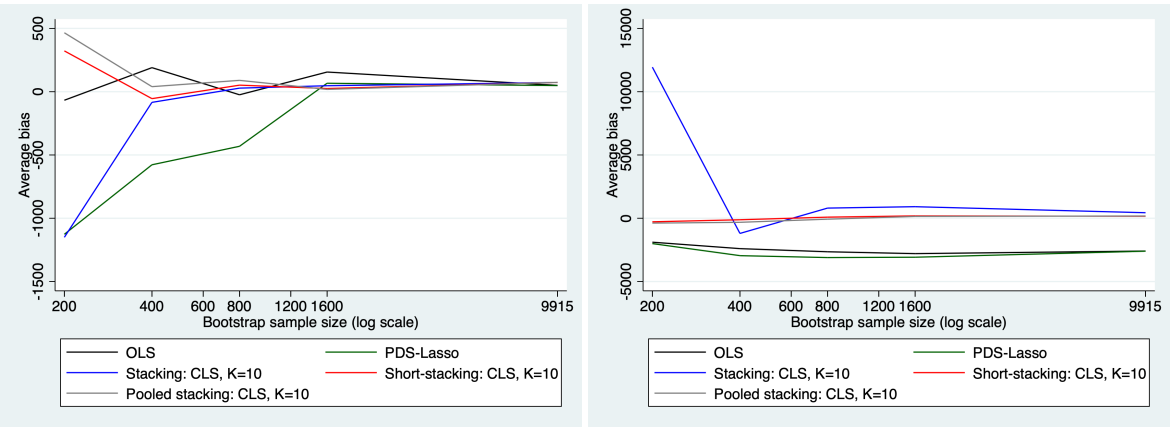
Notes: The table reports the stacking weights corresponding to the DDML short-stacking estimators in Figure 3. Panel A-B use constrained least squares. Panel C-D rely on the single-best final learner. Panel A and C refer to the estimation of $E[Y|X]$; Panel C and D to the estimation of $E[D|X]$. See notes below Table 3 for more information.

In the next step, we expand the set of candidate learners by two types of random forests, two types of gradient-boosted trees and a feed-forward neural net. In principle, widening the set of candidate learners increases robustness to a larger class of unknown confounding structures. We show the results in Panel C, Table 3. When measuring performance based on the difference to the full-sample estimates, we find there are benefits of extending the set of candidate learners for bootstrap sample sizes of $n_b = 800$ or larger. The results are generally comparable across conventional, short and pooled stacking. However, single-best exhibits a lower bias for small bootstrap sample sizes vs. CLS, while pooled stacking with CLS appears to perform worse. The CLS weights reported in

Appendix Table B.2 illustrate how DDML-stacking estimators adapt to the sample size. For example, for smaller sample sizes, a larger weight is put on OLS in the estimation of $E[Y|X]$. In Panel D of Table 3, we report results for each candidate learner individually. DDML-stacking approaches perform better than most individual candidate learners and similar to the best-performing individual learner, which is DDML with CV lasso and QSI controls. In Panel E, we also show results if we reduce the number of folds to $K = 2$. The performance deteriorates drastically for smaller sample sizes, indicating that—while DDML stacking appears competitive for small sample sizes—it is important to increase the number of folds to ensure larger training samples for the CEF estimators.

A drawback of measuring the bias as the difference to the full-sample estimate is that we do not gain insights about convergence to the true parameter. We thus revisit the calibrated simulation exercise from Section 4.1, which allows us to measure the bias as the difference to the true parameter. When the DGP is linear (see Figure 5a), DDML with short-stacking or pooled stacking and using CLS performs overall similarly to OLS. DDML with conventional stacking exhibits relatively large bias with $n_b = 200$. If the true DGP is non-linear, see Figure 5b, OLS and PDS-Lasso are unable to recover the true effect, while DDML with short and pooled stacking and CLS yield reasonably close approximations of the true parameter even for small sample sizes. DDML with conventional stacking is competitive only for larger samples. We provide extensive results for mean bias and coverage rates in Tables B.3–B.5 in the Appendix.

Figure 5: Mean bias for very small sample sizes



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

Notes: The figure shows results from the calibrated simulation in Table 1, but with smaller bootstrap sample sizes. See table notes in Table 1 for more information. Full results for bias and coverage in small samples can be found in Table B.3, B.4 and B.5.

To conclude, the results highlight the risks of relying on inappropriate functional form assumptions. DDML paired with stacking approaches—when combined with a diverse set of candidate learners—imposes weaker conditions on the underlying data-generating process compared to relying on a single pre-selected learner. Short-stacking and pooled stacking outperform conventional stacking in small samples. We conjecture the improvement is due to short and pooled stacking imposing common weights across cross-fitting folds. The use of a common set of weights imposes regularization that is consistent with learner performance being stable across subsamples, which seems like a natural benchmark. For pooled stacking, this additional regularization should reduce weight variance while coming with relatively little cost in terms of bias. Because short-stacking does not make use of the additional train and test splits from conventional stacking, it has more potential to suffer from an additional over-fitting bias. When a small number of learners is considered, this additional bias should be small relative to the variance reduction obtained from not needing to estimate fold-specific stacking weights. The simulation results provide support in favor of this conjecture.

5 Applications

In this section, we use two applications to illustrate how pairing DDML and stacking can increase the robustness of structural parameter estimates to the underlying structure of the data. In the first application, we estimate gaps in citations of articles in top economics journals across different gender compositions among the authors. We condition on the abstract to proxy for the content and quality of the paper and demonstrate that stacking-based DDML is a practical solution to challenging estimation problems using text data. In the second application, we revisit the UK sample of the OECD Skills Survey for Kitagawa-Oaxaca-Binder estimates of the unexplained gender wage gap where we condition on a large set of individual characteristics. Both applications pertain to the literature on 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), and are methodologically also closely related to the broader literature on discriminatory attitudes towards minority groups (e.g., Hangartner, Kopp, and Siegenthaler, 2021).

5.1 Gender gap in citations

This section uses DDML with stacking to estimate a partially linear model applied to average differences in citations of articles published in top-30 economic journals from 1983 to 2020 by the gender composition of the authors. Following Card et al. (2020), we distinguish between papers with (imputed) all-male, all-female, and mixed-gender authorship.[16] Instead of conditioning on hand-coded characteristics such as JEL codes, we leverage the abstract text as a proxy for the topic and quality of the article. Estimating these conditional differences is a challenging statistical problem due to the non-standard nature of text data, and researchers are faced with two key decisions when operationalizing an estimator using text data: how to encode the text data into numerical features, and how to select a suitable learner given the encoded data. Both decisions are ex-ante challenging, but also practically highly relevant as text data is becoming increasingly encountered in economic applications (e.g., Gentzkow and Shapiro, 2010; Ash, Chen, and Ornaghi, 2024; Widmer, Galletta, and Ash, 2023). We show that these decisions can be consequential and that by simultaneously considering different encoding procedures and multiple learners, DDML with stacking provides a simple practical solution to both problems.

In documenting average differences in citations, the analysis presented also contributes to the broader literature on gender biases in academia (e.g., Lundberg and Stearns, 2019; Card et al., 2020; Hengel, 2022). It is well-documented that women are under-represented in academia, especially in senior positions (Ceci et al., 2014; Lundberg and Stearns, 2019). A possible reason for the persistent gap in representation include is that scholarly work produced by women faces more sceptical scrutiny compared to work produced by their male counterparts (Hengel, 2022; Krawczyk and Smyk, 2016). Higher scrutiny could be, for example, reflected at the refereeing stage when a publication decision is made and, as we examine here, after publication when scholarly work is attributed by other scholars through citations (Card et al., 2020; Roberts, Stewart, and Nielsen, 2020; Grossbard, Yilmazer, and Zhang, 2021).

Throughout our analysis, we focus on a descriptive characterization of the average gaps in citations across different gender compositions of the authors as given by θ_0 in the partially linear model of Equation (1) where Y denotes log-citations, D is a two-dimensional vector whose first component is an indicator for all-female authorship and whose second

[16] As we explain below, we impute the gender mix of authors from the authors' names.

component is an indicator for mixed-gender authorship. The vector X collects the content of the abstract and a set of year-of-publication indicators. The two components of θ_0 may thus be interpreted as summarizing the average relative difference in total citations between all-male and all-female authorship, and all-male and mixed-gender authorship, respectively, conditional on the article's year of publication and abstract. Throughout, we make no conditional unconfoundedness assumptions that would be necessary for causal interpretations.

We consider a sample of 27 599 articles that have been published between 1983–2020. The data was sourced from Scopus and is a sub-sample of the data analyzed in Advani et al. (2021), who kindly shared their data with us. For each article, we have a record of the citation count and the authors' names, which we use to infer the authors' gender.[17] In the sample, 6.3% of articles are authored by only female authors and 22.9% have authors from both genders.

Before turning to estimation, the text of the abstract needs to be transformed into a numerical vector. To admit estimation conditional on the content of the abstract, it is necessary to find a representation (referred to as an embedding) of the text that is lower-dimensional but captures its core meaning. An active literature in statistics and computer science provides solutions to this problem, suggesting a large variety of algorithms to construct text embeddings (see the overview in Ash and Hansen, 2023). Thus, in addition to the choice of candidate learner, researchers intent on using text data for their analysis are faced with the additional choice of embedding algorithm. To illustrate how stacking-based DDML can help support this choice, we consider two procedures for encoding the text of the abstract into numerical features: First, we consider a bag-of-word model summarizing the text as (stemmed) word counts (as used in, e.g., Enke, 2020; Esposito et al., 2023). In our data, this results in a 211-dimensional vector of word counts for each abstract. Second, since the bag-of-word approach disregards the word order and context, we construct word embeddings generated by a pre-trained BERT model, a transformer-based large-language model (Devlin et al., 2018). In particular, for each abstract, we

[17] We use the software *Namsor*, which frequently ranks among the best-performing algorithms for gender classification using names (Sebo, 2021; Krstovski, Lu, and Xu, 2023) and is widely used in academic studies (e.g., Bursztyn et al., 2021; Sebo and Schwarz, 2023). In the main specification, we exclude articles of authors whose gender could not be classified with a probability of less than 70%, but we show that results are similar when we apply thresholds of 60% or 90%; see Appendix Figure C.1. Our sample includes 586 articles for which no citation is recorded. These were excluded from the analysis. We also provide results using the number of citations (instead of log-citations) in Appendix Table C.1.

extract the 768-dimensional vector of weights from the last hidden layer of the BERT model that was pre-trained on a large corpus of (uncased) English text data.[18] Instead of embedding individual words, BERT attempts to reconstruct both whole sentences and the context of these sentences, making it particularly suitable to characterize the content of the abstracts. Recently, Bajari et al. (2023) use BERT to construct embeddings of product descriptions on Amazon.com.

The numerical abstract embeddings are then used in several base learners. We consider OLS, PDS lasso and DDML with CV lasso, CV ridge, `XGBoost` (Chen and Guestrin, 2016), random forests and a feed-forward neural net (see table notes for details).[19] The base learners are aggregated by pairing DDML with either conventional stacking or short-stacking, and with either CLS or single-best.[20] The final estimator thus simultaneously combines both text embedding algorithms and machine learning algorithms.

Figure 6 shows estimates of the average relative difference in total citations between all-male and all-female authorship (top-panel) and all-male and mixed-gender authorship (bottom-panel), respectively, for different control specifications and estimators. When we only condition on the publication year, the citation penalty for all-female authorship is close to zero, while there is a large positive effect of +22.6% ($s.e. = 1.7$) for mixed-gender authorship. We next employ PDS lasso to add the abstract text either in the form of word counts or as BERT features. Using the latter, the citation gap increases to −7.4% (2.9) for articles with all-female authorship, while the average relative difference of articles with mixed-gender authorship reduces to +9.2% (1.7). The estimates are qualitatively similar when using word counts instead of BERT features.

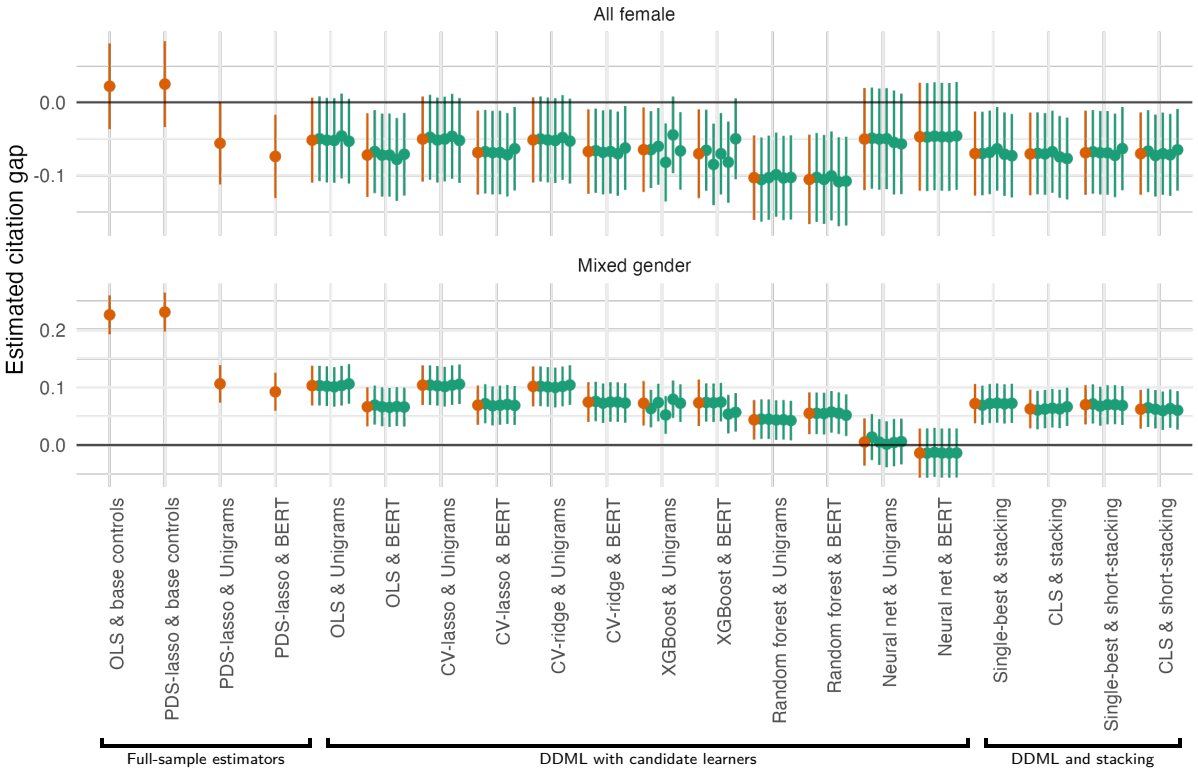
In the figure, we also show five cross-fitting repetitions of pairing DDML with each candidate learner (in green) and the median aggregates over these repetitions (in orange). There are considerable differences across DDML estimators, with the median estimates of the citation gap ranging between −4.7% (3.8) and −10.6 (3.1) for articles with all-female authorship and between −1.4 (2.2) and +10.4 (1.8) for articles with mixed-gender authorship, highlighting that different candidate learner specifications can yield vastly different effect sizes. These stark differences emphasize the need to choose and tune CEF estimators

[18]The model `bert-base-uncased` is freely available from, among others, the Python library `huggingface`.

[19]To reduce the run time, we use regression approaches both for the estimation of $E[Y|X]$ and $E[D|X]$.

[20]We omit pooled stacking from this application since the `R` package `ddml`, which was used for this application, does not currently support pooled stacking.

Figure 6: The citation gap by authors' gender composition



Notes: The figure shows estimates of θ_0 summarizing average relative difference in total citations between all-male and all-female authorship, and all-male and mixed-gender authorship, respectively, conditional on the article's year of publication and abstract. Error-bars show heteroskedasticity-robust 95% confidence intervals. The sample mean (and standard deviation) of citation counts, all female and mixed gender are 99.8 (254.0), 0.062 (0.242) and 0.229 (0.420), respectively. We consider the following estimators: OLS, PDS lasso and DDML with the following candidate learners: OLS, CV ridge, CV lasso, XGBoost (using 500 trees, learning rate of 0.3, maximum depth of 6), random forest (using 500 trees) and feed-forward neural net (early stopping with 15 rounds, 0.5 dropout, 0.1 learning rate, 0.1 validation split, 50 epochs, 500 batch size and 3 hidden layers of size 10). Finally, we pair DDML with either conventional stacking or short-stacking, and with either CLS or single-best as the final learner based on the above candidate learners. Throughout, we use five cross-fitting repetitions, five cross-validation folds and five cross-fitting folds. Results from each cross-fitting replication are illustrated in green, and median aggregates across the cross-fitting replications are shown in orange. The sample includes 29 185 articles published between 1983–2020 in top-30 economics journals. A tabular version is provided in Table C.1. Authors that could not be assigned a gender with less than 70% probability are excluded from the analysis, but we show results based on thresholds of 60% and 90% in Appendix Figure C.1.

carefully. Without thoroughly validating each candidate learner, judging which results are more credible is difficult. Furthermore, it is noteworthy that some candidate learners, specifically those based on XGBoost in this example, exhibit substantial instability across cross-fitting repetitions.

We show results from pairing DDML and stacking approaches on the right-hand side of the same figure. Relative to the DDML estimates based on the individual candidate learners, the stacking approaches yield lower variability over cross-fitting repetitions. All four stacking-based approaches agree on an average relative difference in citations of around −7.0% (2.9) for articles with all-female authorship and suggest a citation advantage of between +6.2 (1.7) and +7.0 (1.7) for articles with mixed-gender authorship.

Table 5: Stacking weights in the gender citation gap application.

	Citations		*All female*		*Mixed gender*	
	Conv.	*Short*	*Conv.*	*Short*	*Conv.*	*Short*
Panel A. Stacking and short-stacking weights						
OLS & Unigrams	0.1	0.069	0.113	0.155	0.135	0.114
OLS & BERT	0.35	0.368	0.09	0.106	0.174	0.196
CV-lasso & Unigrams	0.	0.	0.049	0.013	0.	0.
CV-lasso & BERT	0.119	0.102	0.363	0.396	0.129	0.166
CV-ridge & Unigrams	0.	0.	0.	0.	0.	0.
CV-ridge & BERT	0.	0.	0.361	0.31	0.383	0.329
XGBoost & Unigrams	0.217	0.236	0.008	0.008	0.017	0.028
XGBoost & BERT	0.171	0.174	0.016	0.01	0.033	0.035
Random forest & Unigrams	0.047	0.055	0.02	0.026	0.149	0.151
Random forest & BERT	0.	0.	0.001	0.	0.	0.
Neural net & Unigrams	0.	0.	0.	0.	0.	0.
Neural net & BERT	0.	0.	0.	0.	0.	0.

	Citations	*All female*	*Mixed gender*
Panel B. Mean-squared prediction error			
OLS & Unigrams	1.341	0.058	0.166
OLS & BERT	1.294	0.059	0.166
CV-lasso & Unigrams	1.339	0.058	0.165
CV-lasso & BERT	1.274	0.057	0.161
CV-ridge & Unigrams	1.341	0.058	0.165
CV-ridge & BERT	1.286	0.057	0.161
XGBoost & Unigrams	1.439	0.075	0.196
XGBoost & BERT	1.472	0.068	0.191
Random forest & Unigrams	1.358	0.059	0.167
Random forest & BERT	1.52	0.059	0.169
Neural net & Unigrams	2.082	0.059	0.177
Neural net & BERT	2.207	0.059	0.177

Notes: Panel A shows stacking weights for conventional stacking (labelled 'Conv.') and short-stacking (labelled 'Short') by candidate learners and by variable. Panel B reports the mean-squared prediction error. The final learner is constrained least squares. The stacking weights and mean-squared prediction errors are averaged over cross-fitting repetitions. Treatment variables are an indicator for all-female authors and mixed-gender authors.

Table 5 shows the stacking weights of conventional and short-stacking with constrained least squares as the final learner along with mean-squared prediction errors. The stacking estimators assign small weights to learners exhibiting a relatively large MSPE and

large variability over cross-fitting repetitions. For example, in the CEF estimation of log citations, the neural nets have an MSPE that is around 50% larger than that of other learners. Stacking assigns, as desired, zero weights to the neural nets, whereas OLS leveraging BERT as one of the best-performing learners receives the largest weights. It is noteworthy that the stacking weights often vary markedly across CEFs, highlighting that there is no reason to assume that the same learner is best suited for estimating both $E[Y|X]$ and $E[D|X]$. This insight is especially important since most estimation approaches (including OLS and PDS lasso) impose the same structure for each CEF.

The results on the citation gaps in top economic journals conditional on the content of the abstract are consistent with a citation penalty for all-female authored articles, possibly due to a higher degree of skepticism towards all-female author teams compared to all-male author teams. However, similar to Card et al. (2020) and Maddi and Gingras (2021), the estimates also suggest a conditional citation advantage of articles with mixed-gender authorship.

5.2 Gender gap in wages

The gap in wages between men and women is a central measure of economic gender equality and has been the focus of an extensive empirical literature (see, e.g., the review in Blau and Kahn, 2017). The classic approach to estimating the unexplained gender wage gap relies on a linear version of the Kitagawa-Oaxaca-Binder decomposition (Kitagawa, 1955; Oaxaca, 1973; Blinder, 1973; for an overview, see Fortin, Lemieux, and Firpo, 2011). Several recent articles by Bonaccolto-Töpfer and Briel (2022), Strittmatter and Wunsch (2021), Böheim and Stöllinger (2021) and Bach, Chernozhukov, and Spindler (2023), among others, focus instead on semi-parametric decompositions of the wage gap leveraging more flexible machine learning algorithms. Much of this literature focuses, however, on lasso-based approaches, even though there is no apparent reason to favor sparsity-based approaches over learners relying on other regularization assumptions. In contrast to the recent literature that primarily focuses on lasso-based approaches, we consider a diverse set of candidate learners and aggregate them via stacking.

The parameter of interest in this application is the unexplained gender wage gap, which is the expected difference in wages after conditioning on observed characteristics.

Formally,

$$\theta_0 \equiv E\left[E\left[Y|D=1,X\right]-E\left[Y|D=0,X\right]|D=1\right],$$

where Y denotes the logarithm of wages, D is an indicator equal to one for women, and X is a vector of potentially many individual characteristics. The parameter is well-defined if $P(D=1|X)>0$ with probability 1.[21]

In the absence of functional form assumptions, estimation of θ_0 is a challenging statistical problem due to its dependence on unknown conditional expectation functions that need to be nonparametrically estimated. Analogous to the DDML estimator for the partially linear model outlined in Section 2, we consider estimation of θ_0 via the split-sample analogue of the efficient score function for θ_0[22] – i.e.,

$$\hat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_i(Y_i-\hat{g}_{I^c_{k_i}}(0,X_i))}{\hat{p}_{I^c_{k_i}}}-\frac{\hat{m}_{I^c_{k_i}}(X_i)(1-D_i)(Y_i-\hat{g}_{I^c_{k_i}}(0,X_i))}{\hat{p}_{I^c_{k_i}}(1-\hat{m}_{I^c_{k_i}}(X_i))}\right),$$

where $\hat{g}_{I^c_k}$ and $\hat{m}_{I^c_k}$ are cross-fitted estimators for $g_0(D,X)\equiv E[Y|D,X]$ and $m_0(X)\equiv E[D|X]$, and $\hat{p}_{I^c_k}$ is a cross-fitted estimator of $P(D=1)$.

Following Forshaw et al. (2024), we take the data for this application from the UK sample of the OECD Skills Survey, which was collected in 2011-12 and comes with a rich set of covariates, including age, experience, education, occupation, and industry. The final data includes 4 836 British respondents. We specify three sets of control variables. The *simple* set of controls only includes a selection of essential covariates: age (in levels and squared), years of education, a literacy and numeracy test score, years of tenure in the current job (in levels and squared), education level, hours worked per week, and number of children. The *base* set of controls adds, among others, management level, age of children, and parents' education level.[23] Furthermore, we interact age and tenure with all categorical covariates. The *extended* set of controls comprises all variables and interacts each continuous covariate with each categorical covariate.

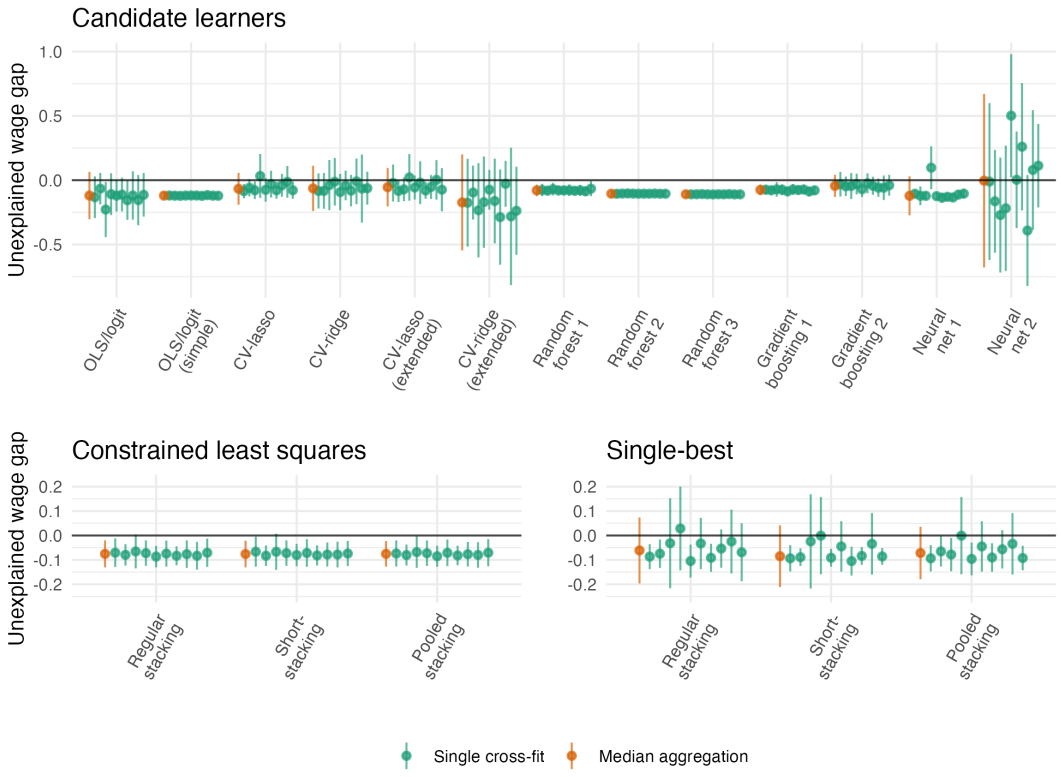
[21] As in the previous section, we focus our analysis on a descriptive parameter of interest and do not make conditional unconfoundedness assumptions that would be necessary for causal interpretations.

[22] See, e.g., equation (5.4) in Chernozhukov et al. (2018).

[23] The base set adds the following variables to the reduced set: area of study, part of larger organization, management position, type of contract, job satisfaction, health status, living with a partner, age of youngest child, immigration age, mother's and father's highest level of education, immigration status of parents, informal job-related education in last 12 months, informal non-job-related education in last 12 months.

We include a diverse set of candidate learners to allow for a high level of flexibility. We employ regression approaches for the estimation of the CEF of log wages (i.e., $E[Y|D,X]$) and classification approaches for the CEF estimation of gender (i.e., $E[D|X]$). Our candidate learners are linear (or logistic) regression with the simple and base set of controls; linear (or logistic) CV-lasso and CV-ridge with the base and extended set of controls; three random forests with 500 regression (or classification) trees and minimum leaf sizes of 1, 50 and 100; two types of gradient-boosted regression (or classification) trees with and without early stopping (500 trees, maximum depth of 3); two feed-forward neural nets with hidden layer sizes of $(40, 20, 1, 20, 50)$ and $(30, 30, 30)$ and early stopping. Finally, we aggregate the candidate learners via conventional, short and pooled stacking, using either CLS or single-best as the final learner.

Figure 7: Unexplained gender wage gap



Notes: The figure reports DDML estimates of the unexplained gender wage gap based on several different learners. 95% heteroskedasticity-robust confidence intervals are shown. The sample mean (and standard deviation) of log earnings and gender are 2.768 (0.579) and 0.583 (0.493), respectively. The candidate learners are OLS (for the outcome equation) and logit (for the propensity scores) with the base and simple set of controls; CV-lasso and CV-ridge with the base and extended set of controls; three random forests with 500 trees and minimum leaf sizes of 1, 50 and 100; two gradient-boosted trees with and without early stopping (500 trees, with and without early stopping after 10 iterations, maximum depth of 3); two feed-forward neural nets with hidden layer sizes of $(40, 20, 1, 20, 50)$ and $(30, 30, 30)$ and early stopping. We report results for the individual base learners in the top panel. In the bottom panels, we show DDML paired with conventional, short and pooled stacking based on the base learners and with either CLS (left panel) or single-best (right panel) as the final learner. We use 10 cross-fitting folds and 10 cross-fitting repetitions. Results from each cross-fitting replication are illustrated in green, and median aggregates across the cross-fitting replications are shown in orange. A tabular version is provided in Table D.4-D.5.

Figure 7 reports results for individual candidate learners (on the top) and stacking approaches (on the bottom). We show results from 10 cross-fitting repetitions (in green) and the median aggregates (in orange). We again find that some candidate learners exhibit substantial variability over cross-fitting repetitions, which is also reflected in the large median-aggregate standard errors. The variability is especially large for CV-ridge with the extended set of controls and the neural nets, which are the candidate learners exhibiting the largest MSE (see Appendix Table D.1). The stacking results are, in contrast, relatively stable over cross-fit repetitions when using CLS as the final learner. Interestingly, the results when relying on single-best as the final learner are noticeably more variable than when using CLS, indicating that a combination of candidate learners seems to better fit the data than a single learner. The stacking weights and MSE in Appendix Table D.1 confirm that there is no single candidate learner dominating the others. The instability of the single-best final learner is reflected in the stacking standard errors. Given this potential for instability of choosing a single candidate learner, we recommend favoring constrained least squares over single-best if one is not confident that one of the chosen learners will be significantly better than the rest, and thus stably selected, which seems likely to be the most common setting in practice.

6 Conclusion

This article assesses the performance of DDML estimators in realistic settings using applications and simulation studies calibrated to real economic data. We highlight that estimators of structural parameters based on single pre-selected (machine) learners can be highly sensitive to the underlying structure of the data and/or poor tuning, and we show that pairing DDML with stacking can help alleviate these concerns, provided that a sufficiently diverse set of candidate learners is considered.

We discuss pairing DDML with conventional stacking but also suggest two alternative stacking approaches: Short-stacking, which substantially reduces the computational burden by leveraging the cross-fitting naturally arising in the computation of DDML estimates, and pooled stacking, which decreases the variance of the stacking estimator by imposing common stacking weights over cross-fitting folds. In our simulations, both strategies are competitive with conventional stacking in settings with large and moder-

ate sample sizes and are better in small samples. The advantages of short-stacking are particularly worth highlighting, given its substantially lower computational cost.

A key advantage of the DDML-stacking approach is that it accommodates both traditional parametric and nonparametric specifications by allowing simultaneous consideration of, for example, OLS with several sets of controls, sparsity-based learners, tree-based ensembles and neural networks. In this sense, researchers are not forcibly deviating from standard (often linear) specifications unless the data suggests there is reason to. While machine-learning-based causal methods may yield fundamentally different results from linear regression only in specific examples, the additional robustness to unexpected structures in the data thus seems to come at relatively little cost.

References

Advani, Arun, Elliott Ash, David Cai, and Imran Rasul (2021). *Race-related research in economics and other social sciences*. Discussion Paper 16115. CEPR.

Ahrens, Achim, Christian B Hansen, and Mark E Schaffer (2018). *PDSLASSO: Stata module for post-selection and post-regularization OLS or IV estimation and inference*. URL: `https://ideas.repec.org/c/boc/bocode/s458459.html`.

Ahrens, Achim, Christian B. Hansen, and Mark E. Schaffer (2023). "pystacked: Stacking generalization and machine learning in Stata". *The Stata Journal* 23.4, pp. 909–931.

Ahrens, Achim, Christian B. Hansen, Mark E. Schaffer, and Thomas Wiemann (2024). "ddml: Double/debiased machine learning in Stata". *The Stata Journal* 24.1, pp. 3–45.

Angrist, Joshua D. (2022). "Empirical Strategies in Economics: Illuminating the Path From Cause to Effect". *Econometrica* 90.6, pp. 2509–2539.

Angrist, Joshua D and Brigham Frandsen (2022). "Machine labor". *Journal of Labor Economics* 40.S1, S97–S140.

Angrist, Joshua D and Alan B Krueger (1999). "Empirical strategies in labor economics". In: *Handbook of Labor Economics*. Vol. 3. Elsevier, pp. 1277–1366.

Ash, Elliott, Daniel L. Chen, and Arianna Ornaghi (2024). "Gender Attitudes in the Judiciary: Evidence from US Circuit Courts". *American Economic Journal: Applied Economics* 16.1, 314–50.

Ash, Elliott and Stephen Hansen (2023). "Text Algorithms in Economics". *Annual Review of Economics* 15.1, annurev–economics–082222–074352.

Athey, Susan, Julie Tibshirani, and Stefan Wager (2019). "Generalized random forests". *Annals of Statistics* 47.2, pp. 1148–1178.

Bach, Philipp, Victor Chernozhukov, and Martin Spindler (2023). "Heterogeneity in the US gender wage gap". *Journal of the Royal Statistical Society Series A: Statistics in Society* 187.1, pp. 209–230.

Bajari, Patrick et al. (2023). "Hedonic prices and quality adjusted price indices powered by AI". *arXiv preprint arXiv:2305.00044*.

Bates, J.M. and C.W.J. Granger (1969). "The combination of forecasts". *Journal of the Operational Research Society* 20.4, pp. 451–468.

Belloni, A, V Chernozhukov, I Fernández-Val, and C Hansen (2017). "Program Evaluation and Causal Inference With High-Dimensional Data". *Econometrica* 85.1, pp. 233–298.

Belloni, Alexandre, D Chen, Victor Chernozhukov, and Christian Hansen (2012). "Sparse Models and Methods for Optimal Instruments With an Application to Eminent Domain". *Econometrica* 80.6, pp. 2369–2429.

Belloni, Alexandre, Victor Chernozhukov, and Christian Hansen (2014). "Inference on treatment effects after selection among high-dimensional controls". *Review of Economic Studies* 81, pp. 608–650.

Bickel, Peter J, Ya'acov Ritov, and Alexandre B Tsybakov (2009). "Simultaneous analysis of Lasso and Dantzig selector". *Annals of Statistics* 37.4, pp. 1705–1732.

Blandhol, Christine, John Bonney, Magne Mogstad, and Alexander Torgovitsky (2022). "When is TSLS Actually LATE?" *BFI Working Paper* 2022-16.

Blau, Francine D. and Lawrence M. Kahn (2017). "The Gender Wage Gap: Extent, Trends, and Explanations". *Journal of Economic Literature* 55.3, pp. 789–865.

Blinder, Alan (1973). "Wage Discrimination: Reduced Form and Structural Estimates". *Journal of Human Resources* 8, pp. 436–455.

Böheim, René and Philipp Stöllinger (2021). "Decomposition of the gender wage gap using the LASSO estimator". *Applied Economics Letters* 28.10, pp. 817–828.

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

Breiman, Leo (1996). "Stacked regressions". *Machine Learning* 24.1, pp. 49–64.

Buitinck, Lars et al. (2013). "API design for machine learning software: experiences from the scikit-learn project". In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122.

Bursztyn, Leonardo, Thomas Chaney, Tarek Alexander Hassan, and Aakaash Rao (2021). *The Immigrant Next Door: Long-Term Contact, Generosity, and Prejudice*. Working Paper 28448. National Bureau of Economic Research. URL: http://www.nber.org/papers/w28448.

Callaway, Brantly and Pedro HC Sant'Anna (2021). "Difference-in-differences with multiple time periods". *Journal of Econometrics* 225.2, pp. 200–230.

Card, David, Stefano DellaVigna, Patricia Funk, and Nagore Iriberri (2020). "Are Referees and Editors in Economics Gender Neutral?" *The Quarterly Journal of Economics* 135.1, pp. 269–327.

Card, David and A. Abigail Payne (2021). "High School Choices and the Gender Gap in Stem". *Economic Inquiry* 59.1, pp. 9–28.

Ceci, Stephen J., Donna K. Ginther, Shulamit Kahn, and Wendy M. Williams (2014). "Women in Academic Science: A Changing Landscape". *Psychological Science in the Public Interest: A Journal of the American Psychological Society* 15.3, pp. 75–141.

Chang, Neng-Chieh (2020). "Double/debiased machine learning for difference-in-differences models". *The Econometrics Journal* 23.2, pp. 177–191.

Chen, Tianqi and Carlos Guestrin (2016). "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Chernozhukov, Victor, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins (2018). "Double/debiased machine learning for treatment and structural parameters". *The Econometrics Journal* 21.1, pp. C1–C68.

Clemen, Robert T. (1989). "Combining forecasts: A review and annotated bibliography". *International Journal of Forecasting* 5.4, pp. 559–583.

Clydec, Merlise and Edwin S Iversen (2013). "Bayesian model averaging in the M-open framework". In: *Bayesian Theory and Applications*. Ed. by Paul Damien, Petros Dellaportas, Nicholas G. Polson, and David A. Stephens. Oxford University Press, p. 0. URL: https://doi.org/10.1093/acprof:oso/9780199695607.003.0024.

Colangelo, Kyle and Ying-Ying Lee (2023). *Double Debiased Machine Learning Nonparametric Inference with Continuous Treatments*. URL: https://arxiv.org/abs/2004.03036.

Crane, Dwight B. and James R. Crotty (1967). "A Two-Stage Forecasting Model: Exponential Smoothing and Multiple Regression". *Management Science* 13.8, B–501–B–507.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". *arXiv preprint arXiv:1810.04805*.

Dhar, Diva, Tarun Jain, and Seema Jayachandran (2022). "Reshaping adolescents' gender attitudes: Evidence from a school-based experiment in India". *American Economic Review* 112.3, pp. 899–927.

Eberhardt, Markus, Giovanni Facchini, and Valeria Rueda (2023). "Gender Differences in Reference Letters: Evidence from the Economics Job Market". *The Economic Journal* 133.655, pp. 2676–2708.

Enke, Benjamin (2020). "Moral Values and Voting". *Journal of Political Economy* 128.10, pp. 3679–3729.

Esposito, Elena, Tiziano Rotesi, Alessandro Saia, and Mathias Thoenig (2023). "Reconciliation narratives: The birth of a nation after the US civil war". *American Economic Review* 113.6, pp. 1461–1504.

Farrell, Max H, Tengyuan Liang, and Sanjog Misra (2021). "Deep neural networks for estimation and inference". *Econometrica* 89.1, pp. 181–213.

Forshaw, Rachel, Vsevolod Iakovlev, Mark E. Schaffer, and Cristina Tealdi (2024). "Using machine learning methods to estimate the gender wage gap". In: *Machine Learning for Econometrics and Related Topics*. Ed. by Vladik Kreinovich, Songsak Sriboonchitta, and Woraphon Yamaka. Springer Cham. URL: https://doi.org/10.1007/978-3-031-43601-7.

Fortin, Nicole, Thomas Lemieux, and Sergio Firpo (2011). "Chapter 1 - Decomposition Methods in Economics". In: *Handbook of Labor Economics*. Ed. by Orley Ashenfelter and David Card. Vol. 4. Elsevier, pp. 1–102. URL: https://www.sciencedirect.com/science/article/pii/S0169721811004072.

Gentzkow, Matthew and Jesse M. Shapiro (2010). "What drives media slant? Evidence from U.S. daily newspapers". *Econometrica* 78.1, pp. 35–71.

Giannone, Domenico, Michele Lenza, and Giorgio E Primiceri (2021). "Economic predictions with big data: The illusion of sparsity". *Econometrica* 89.5, pp. 2409–2437.

Gilchrist, Duncan Sheppard and Emily Glassberg Sands (2016). "Something to talk about: Social spillovers in movie consumption". *Journal of Political Economy* 124.5, pp. 1339–1382.

Goller, Daniel, Michael Lechner, Andreas Moczall, and Joachim Wolff (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, p. 101855.

Grossbard, Shoshana, Tansel Yilmazer, and Lingrui Zhang (2021). "The gender gap in citations of articles published in two demographic economics journals". *Review of Economics of the Household* 19.3, pp. 677–697.

Hangartner, Dominik, Daniel Kopp, and Michael Siegenthaler (2021). "Monitoring hiring discrimination through online recruitment platforms". *Nature* 589.7843, pp. 572–576.

Hansen, Bruce E. and Jeffrey S. Racine (2012). "Jackknife model averaging". *Journal of Econometrics* 167.1, pp. 38–46.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. 2nd ed. New York: Springer-Verlag.

Hengel, Erin (2022). "Publishing While Female: are Women Held to Higher Standards? Evidence from Peer Review". *The Economic Journal* 132.648, pp. 2951–2991.

Kitagawa, Evelyn M. (1955). "Components of a Difference Between Two Rates". *Journal of the American Statistical Association* 50.272, pp. 1168–1194.

Kolesár, Michal, Ulrich K. Müller, and Sebastian T. Roelsgaard (2023). *The Fragility of Sparsity*. URL: `http://arxiv.org/abs/2311.02299`.

Krawczyk, Michał and Magdalena Smyk (2016). "Author's gender affects rating of academic articles: Evidence from an incentivized, deception-free laboratory experiment". *European Economic Review*. Social identity and discrimination 90, pp. 326–335.

Krstovski, Kriste, Yao Lu, and Ye Xu (2023). *Inferring gender from name: a large scale performance evaluation study*. URL: `https://arxiv.org/abs/2308.12381`.

Laan, Mark J. Van der, Eric C Polley, and Alan E. Hubbard (2007). "Super Learner". *Statistical Applications in Genetics and Molecular Biology* 6.1.

Laan, Mark J. van der and Sandrine Dudoit (2003). "Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples". In: vol. 130. U.C. Berkeley Division of Biostatistics Working Paper Series.

Laan, Mark J Van der and Sherri Rose (2011). *Targeted learning: causal inference for observational and experimental data.* Vol. 4. New York: Springer.

Le, Tri and Bertrand Clarke (2017). "A Bayes Interpretation of Stacking for M-Complete and M-Open Settings". *Bayesian Analysis* 12.3, pp. 807–829.

Lundberg, Shelly and Jenna Stearns (2019). "Women in Economics: Stalled Progress". *Journal of Economic Perspectives* 33.1, pp. 3–22.

Luo, Ye, Martin Spindler, and Jannis Kück (2022). "High-Dimensional L_2 Boosting: Rate of Convergence". *arXiv preprint arXiv:1602.08927.*

Maddi, Abdelghani and Yves Gingras (2021). "Gender diversity in research teams and citation impact in economics and management". *Journal of Economic Surveys* 35.5, pp. 1381–1404.

Oaxaca, Ronald (1973). "Male-Female Wage Differentials in Urban Labor Markets"". *International Economic Review* 14, pp. 693–709.

Poterba, James M, Steven F Venti, and David A Wise (1995). "Do 401 (k) contributions crowd out other personal saving?" *Journal of Public Economics* 58.1, pp. 1–32.

Roberts, Margaret E., Brandon M. Stewart, and Richard A. Nielsen (2020). "Adjusting for Confounding with Text Matching". *American Journal of Political Science* 64.4, pp. 887–903.

Schmidt-Hieber, Johannes (2020). "Nonparametric regression using deep neural networks with ReLU activation function". *Annals of Statistics* 48.4, pp. 1875–1897.

Sebo, Paul (2021). "Performance of gender detection tools: a comparative study of name-to-gender inference services". *Journal of the Medical Library Association : JMLA* 109.3, pp. 414–421.

Sebo, Paul and Joëlle Schwarz (2023). "The level of the gender gap in academic publishing varies by country and region of affiliation: A cross-sectional study of articles published in general medical journals". *PLOS ONE* 18.9, e0291837.

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

Steel, Mark F. J. (2020). "Model Averaging and Its Use in Economics". *Journal of Economic Literature* 58.3, pp. 644–719.

Strittmatter, Anthony and Conny Wunsch (2021). *The Gender Pay Gap Revisited with Big Data: Do Methodological Choices Matter?* URL: https://arxiv.org/abs/2102.09207.

Timmermann, Allan (2006). "Chapter 4 Forecast Combinations". In: *Handbook of Economic Forecasting*. Ed. by G. Elliott, C. W. J. Granger, and A. Timmermann. Vol. 1. Elsevier, pp. 135–196. URL: https://www.sciencedirect.com/science/article/pii/S1574070605010049.

Wager, Stefan and Susan Athey (2018). "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests". *Journal of the American Statistical Association* 113.523, pp. 1228–1242.

Wager, Stefan and Guenther Walther (2016). "Adaptive concentration of regression trees, with application to random forests". *arXiv preprint arXiv:1503.06388*.

Wang, Xiaoqian, Rob J. Hyndman, Feng Li, and Yanfei Kan (2023). "Forecast combinations: An over 50-year review". *International Journal of Forecasting* 39, pp. 1518–1547.

Widmer, Philine, Sergio Galletta, and Elliott Ash (2023). *Media Slant is Contagious*. URL: https://arxiv.org/abs/2202.07269.

Wiemann, Thomas, Achim Ahrens, Christian B Hansen, and Mark E Schaffer (2024). *ddml: Double/Debiased Machine Learning*. URL: https://github.com/thomaswiemann/ddml.

Wolpert, David H. (1992). "Stacked generalization". *Neural Networks* 5.2, pp. 241–259.

Wolpert, David H (1996). "The lack of a priori distinctions between learning algorithms". *Neural computation* 8.7, pp. 1341–1390.

Wüthrich, Kaspar and Ying Zhu (2023). "Omitted variable bias of Lasso-based inference methods: A finite sample analysis". *Review of Economics and Statistics* 105.4, pp. 982–997.

Supplementary material

A The benefits of pairing DDML and stacking

Table A.1: Mean-squared prediction error

	$n_b = 9\,915$		$n_b = 99\,150$	
Panel (A): Linear DGP	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$
Candidate learners				
OLS	3.093	0.200	3.088	0.200
Lasso with CV (2nd order poly)	3.095	0.200	3.088	0.200
Ridge with CV (2nd order poly)	3.100	0.200	3.089	0.200
Lasso with CV (10th order poly)	3.298	0.202	3.095	0.200
Ridge with CV (10th order poly)	3.423	0.206	3.093	0.200
Random forest (low regularization)	3.613	0.233	3.698	0.239
Random forest (high regularization)	3.183	0.205	3.197	0.207
Gradient boosting (low regularization)	3.131	0.201	3.102	0.200
Gradient boosting (high regularization)	3.151	0.201	3.137	0.201
Neural net	3.227	0.204	3.153	0.200
Panel (B): Non-Linear DGP	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$
Candidate learners				
OLS	3.681	0.203	3.672	0.203
Lasso with CV (2nd order poly)	3.480	0.201	3.449	0.200
Ridge with CV (2nd order poly)	3.479	0.201	3.449	0.200
Lasso with CV (10th order poly)	6.161	0.223	3.422	0.200
Ridge with CV (10th order poly)	7.431	0.230	3.424	0.200
Random forest (low regularization)	3.789	0.231	3.515	0.236
Random forest (high regularization)	3.588	0.204	3.251	0.205
Gradient boosting (low regularization)	3.345	0.200	3.095	0.198
Gradient boosting (high regularization)	3.399	0.200	3.216	0.199
Neural net	3.694	0.205	3.510	0.200

Notes: The table shows the mean-squared prediction error of each candidate learner from the simulation example in Section 4.1. The bootstrap sample size is $n_b = 9\,915$ or $99\,150$. Results are based on 1 000 replications. See Table 1 for more information.

Table A.2: Bias in the Linear and Non-Linear DGP

	Panel (A): Linear DGP				*Panel (B): Non-linear DGP*			
	$n_b = 9\,915$		99 150		$n_b = 9\,915$		99 150	
	Bias	s.e.	Bias	s.e.	Bias	s.e.	Bias	s.e.
Full sample:								
OLS	49.9	39.6	-6.8	12.6	-2588.9	46.5	-2632.3	14.8
PDS-Lasso	48.4	39.6	-4.2	12.6	-2598.7	46.5	-2631.6	14.8
DDML methods:								
Candidate learners								
OLS	46.2	39.7	-6.9	12.6	-2613.0	47.5	-2635.4	14.8
Lasso with CV (2nd order poly)	50.9	39.7	-6.2	12.6	703.7	44.5	718.5	13.5
Ridge with CV (2nd order poly)	48.2	39.9	-6.9	12.6	767.4	44.6	729.3	13.5
Lasso with CV (10th order poly)	248.1	266.6	55.9	12.6	-4109.0	1325.8	7.4	22.5
Ridge with CV (10th order poly)	1230.1	401.8	31.6	12.6	-5126.2	1713.9	9.6	23.3
Random forest (low regularization)	-74.7	48.6	-25.2	16.0	-96.1	48.5	-37.5	15.4
Random forest (high regularization)	69.1	41.7	-23.5	13.2	-159.7	43.5	-4.2	13.0
Gradient boosting (low regularization)	12.1	40.1	-24.2	12.6	8.5	42.7	30.9	12.4
Gradient boosting (high regularization)	114.8	39.8	66.9	12.5	162.0	42.1	200.1	12.6
Neural net	394.2	43.1	9.1	13.6	-601.3	46.5	-131.9	14.3
Stacking approaches								
Stacking: CLS	42.8	40.2	-7.5	12.6	133.9	195.6	37.8	12.4
Stacking: Average	107.7	51.2	-6.5	12.6	94.0	110.3	72.3	12.7
Stacking: OLS	-129.3	96.2	-9.4	14.9	-204.8	405.2	17.5	17.8
Stacking: Single-best	43.7	39.9	-8.6	12.6	-121.9	272.9	30.9	12.4
Short-stacking: CLS	45.0	39.7	-7.0	12.6	162.7	42.0	33.6	12.4
Short-stacking: Average	107.7	51.2	-6.5	12.6	94.0	110.3	72.3	12.7
Short-stacking: OLS	37.6	39.6	-7.8	12.6	123.6	41.7	29.5	12.3
Short-stacking: Single-best	44.4	39.7	-8.3	12.6	71.7	42.6	30.9	12.4
Pooled stacking: CLS	58.6	41.4	-7.1	12.6	209.8	63.7	37.5	12.4
Pooled stacking: Average	107.7	51.2	-6.5	12.6	94.0	110.3	72.3	12.7
Pooled stacking: OLS	46.5	47.8	-7.8	12.6	234.5	124.9	30.6	12.3
Pooled stacking: Single-best	46.9	39.7	-8.3	12.6	103.3	45.8	30.9	12.4

Notes: The table reports mean bias and associated standard errors (s.e.) for the listed estimators from the simulation example in Section 4.1. Results are based on 1 000 replications. See Table 1 for more information.

Table A.3: Average stacking weights using OLS as the final learner

	Stacking		Pooled stacking		Short-stacking	
	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$
Panel (A): Linear DGP and $n_b = 9,915$						
OLS	1.161	0.932	1.014	0.815	0.923	0.686
Lasso with CV (2nd order poly)	0.011	0.034	0.066	0.085	0.154	0.079
Ridge with CV (2nd order poly)	–0.157	–0.187	–0.091	–0.137	–0.080	–0.080
Lasso with CV (10th order poly)	–0.046	0.076	–0.027	0.069	–0.041	0.122
Ridge with CV (10th order poly)	0.	–0.006	0.	0.003	0.017	0.022
Random forest (low regularization)	0.001	–0.005	0.001	–0.006	0.003	–0.006
Random forest (high regularization)	–0.014	–0.006	–0.010	–0.002	–0.011	0.008
Gradient boosting (low regularization)	–0.060	–0.108	–0.045	–0.096	–0.039	–0.094
Gradient boosting (high regularization)	0.096	0.260	0.090	0.259	0.076	0.257
Neural net	–195.245	0.009	0.008	0.034	0.004	0.033
Panel (B): Linear DGP and $n_b = 99,150$						
OLS	1.292	0.935	1.130	0.769	1.008	0.677
Lasso with CV (2nd order poly)	–0.103	–0.497	–0.036	–0.378	0.033	–0.398
Ridge with CV (2nd order poly)	–0.170	–0.053	–0.093	–0.019	–0.044	0.056
Lasso with CV (10th order poly)	0.065	0.407	0.023	0.368	0.009	0.335
Ridge with CV (10th order poly)	–0.071	–0.118	–0.022	–0.068	–0.003	–0.034
Random forest (low regularization)	–0.000	–0.001	–0.000	–0.001	0.	–0.001
Random forest (high regularization)	–0.001	–0.002	–0.000	–0.001	–0.000	–0.001
Gradient boosting (low regularization)	–0.034	0.158	–0.021	0.177	–0.020	0.224
Gradient boosting (high regularization)	0.034	0.072	0.027	0.058	0.022	0.020
Neural net	–12.267	0.110	–0.005	0.108	–0.003	0.133
Panel (C): Non-Linear DGP and $n_b = 9,915$						
OLS	0.013	0.050	0.004	0.038	–0.048	0.022
Lasso with CV (2nd order poly)	–0.118	–0.319	–0.166	–0.212	0.033	–0.347
Ridge with CV (2nd order poly)	0.365	0.528	0.427	0.457	0.146	0.548
Lasso with CV (10th order poly)	–0.014	0.126	0.015	0.118	0.091	0.088
Ridge with CV (10th order poly)	0.060	0.028	0.040	–0.003	–0.011	0.055
Random forest (low regularization)	0.052	–0.016	0.056	–0.017	0.057	–0.016
Random forest (high regularization)	–0.098	0.059	–0.099	0.065	–0.110	0.064
Gradient boosting (low regularization)	1.096	0.029	1.133	0.059	1.358	0.159
Gradient boosting (high regularization)	–0.531	0.526	–0.586	0.508	–0.716	0.426
Neural net	–72.340	0.027	0.121	0.046	0.120	0.054
Panel (D): Non-Linear DGP and $n_b = 99,150$						
OLS	–0.015	0.009	–0.017	0.008	0.	0.009
Lasso with CV (2nd order poly)	0.089	–0.806	0.100	–0.802	–0.026	–0.721
Ridge with CV (2nd order poly)	–0.171	0.792	–0.177	0.789	–0.053	0.696
Lasso with CV (10th order poly)	0.285	–0.247	0.229	–0.243	0.358	–0.325
Ridge with CV (10th order poly)	–0.338	0.320	–0.282	0.315	–0.425	0.383
Random forest (low regularization)	0.142	–0.011	0.142	–0.011	0.168	–0.013
Random forest (high regularization)	–0.111	0.069	–0.108	0.070	–0.125	0.079
Gradient boosting (low regularization)	2.368	1.233	2.356	1.236	2.353	1.332
Gradient boosting (high regularization)	–1.443	–0.403	–1.435	–0.406	–1.424	–0.498
Neural net	–45.762	0.030	0.080	0.033	0.066	0.040

Notes: The table shows the (average) stacking weights of each candidate learner for conventional stacking, pooled stacking and short-stacking using OLS as the final learner from the simulation example in Section 4.1. The bootstrap sample size is denoted by n_b. Results are based on 1 000 replications. See Table 1 for more information.

Table A.4: Average stacking weights using single-best

	Stacking		Pooled stacking		Short-stacking	
	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$	$E[Y\|X]$	$E[D\|X]$
Panel (A): Linear DGP and $n_b = 9,915$						
OLS	0.812	0.645	0.889	0.750	0.821	0.650
Lasso with CV (2nd order poly)	0.163	0.274	0.108	0.234	0.150	0.266
Ridge with CV (2nd order poly)	0.021	0.016	0.003	0.003	0.028	0.027
Lasso with CV (10th order poly)	0.002	0.048	0.	0.011	0.001	0.034
Ridge with CV (10th order poly)	0.002	0.010	0.	0.001	0.	0.023
Random forest (low regularization)	0.	0.	0.	0.	0.	0.
Random forest (high regularization)	0.	0.	0.	0.	0.	0.
Gradient boosting (low regularization)	0.	0.003	0.	0.001	0.	0.
Gradient boosting (high regularization)	0.	0.003	0.	0.	0.	0.
Neural net	0.	0.	0.	0.	0.	0.
Panel (B): Linear DGP and $n_b = 99,150$						
OLS	0.895	0.292	0.960	0.288	0.880	0.181
Lasso with CV (2nd order poly)	0.087	0.093	0.037	0.073	0.078	0.059
Ridge with CV (2nd order poly)	0.018	0.016	0.003	0.003	0.042	0.017
Lasso with CV (10th order poly)	0.	0.186	0.	0.172	0.	0.151
Ridge with CV (10th order poly)	0.	0.410	0.	0.464	0.	0.591
Random forest (low regularization)	0.	0.	0.	0.	0.	0.
Random forest (high regularization)	0.	0.	0.	0.	0.	0.
Gradient boosting (low regularization)	0.	0.002	0.	0.	0.	0.001
Gradient boosting (high regularization)	0.	0.	0.	0.	0.	0.
Neural net	0.	0.	0.	0.	0.	0.
Panel (C): Non-linear DGP and $n_b = 9,915$						
OLS	0.	0.	0.	0.	0.	0.
Lasso with CV (2nd order poly)	0.093	0.149	0.066	0.144	0.054	0.081
Ridge with CV (2nd order poly)	0.132	0.124	0.126	0.106	0.058	0.093
Lasso with CV (10th order poly)	0.072	0.058	0.040	0.028	0.039	0.026
Ridge with CV (10th order poly)	0.019	0.041	0.007	0.015	0.005	0.037
Random forest (low regularization)	0.	0.	0.	0.	0.	0.
Random forest (high regularization)	0.002	0.001	0.	0.	0.002	0.
Gradient boosting (low regularization)	0.673	0.357	0.759	0.403	0.832	0.623
Gradient boosting (high regularization)	0.009	0.268	0.002	0.304	0.008	0.140
Neural net	0.002	0.002	0.	0.	0.002	0.
Panel (D): Non-linear DGP and $n_b = 99,150$						
OLS	0.	0.	0.	0.	0.	0.
Lasso with CV (2nd order poly)	0.	0.	0.	0.	0.	0.
Ridge with CV (2nd order poly)	0.	0.	0.	0.	0.	0.
Lasso with CV (10th order poly)	0.	0.	0.	0.	0.	0.
Ridge with CV (10th order poly)	0.	0.	0.	0.	0.	0.
Random forest (low regularization)	0.	0.	0.	0.	0.	0.
Random forest (high regularization)	0.	0.	0.	0.	0.	0.
Gradient boosting (low regularization)	1.	1.	1.	1.	1.	1.
Gradient boosting (high regularization)	0.	0.	0.	0.	0.	0.
Neural net	0.	0.	0.	0.	0.	0.

Notes: The table shows the (average) rates at which each candidate learner is selected by the single-best final learner when using conventional stacking, pooled stacking and short-stacking. The bootstrap sample size is denoted by n_b. Results are based on 1 000 replications. See Table 1 for more information.

Table A.5: Computational time of DDML with conventional and short-stacking

Folds K	Obs.	DDML Stacking Conv.	DDML Stacking Short	OLS	PDS lasso	Ratio
2	200	14.31	3.81	0.0045	0.0461	0.2661
	400	14.93	4.05	0.0047	0.0492	0.2711
	800	18.36	4.95	0.0050	0.0518	0.2696
	1600	26.82	7.07	0.0056	0.0595	0.2636
	9915	138.39	34.35	0.0116	0.1488	0.2482
	99150	2687.07	620.98	0.1013	1.4431	0.2311
5	200	35.77	8.28	0.0045	0.0573	0.2315
	400	41.83	9.79	0.0047	0.0589	0.2342
	800	56.38	13.47	0.0049	0.0624	0.2388
	1600	91.76	21.34	0.0056	0.0711	0.2326
	9915	589.14	136.35	0.0110	0.1508	0.2314
10	200	72.79	16.19	0.0046	0.0423	0.2224
	400	85.87	19.27	0.0046	0.0524	0.2244
	800	119.48	27.80	0.0049	0.0468	0.2327
	1600	197.85	45.59	0.0054	0.0618	0.2304
	9915	1364.07	313.84	0.0113	0.1426	0.2301

Notes: The table reports the computational time in seconds of DDML paired with conventional stacking ('Conv.') or short-stacking ('Short') as implemented in Ahrens et al. (2024), OLS as implemented in Stata's `regress`, post-double-selection lasso as implemented in `pdslasso` (Ahrens, Hansen, and Schaffer, 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). The computations were performed on the high-performance cluster of the ETH Zurich. Each instance used a single core of an AMD EPYC processor with 2.25-2.6GHz (nominal)/3.3-3.5 GHz (peak) and 4GB RAM.

B DDML and stacking in very small samples

Table B.1: Estimates based on the full sample ($N = 9\,915$).

Estimator	*Estimate*
Panel A. No sample splitting	
OLS TWI	6751.907
OLS QSI	5988.413
Post double Lasso TWI c=0.5	6562.923
Post double Lasso QSI c=0.5	5648.14
Post double Lasso TWI c=1	6630.751
Post double Lasso QSI c=1	4646.575
Post double Lasso TWI c=1.5	7474.508
Post double Lasso QSI c=1.5	4472.324
Panel B. DDML with candidate learners	
Neural net	6433.092
OLS	6463.73
Lasso with CV (TWI)	6780.161
Ridge with CV (TWI)	6760.134
Lasso with CV (QSI)	5722.624
Ridge with CV (QSI)	5995.346
Random forest (low regularization)	6089.389
Random forest (high regularization)	6552.221
Gradient boosting (low regularization)	7003.373
Gradient boosting (high regularization)	7992.538
Panel C. DDML with stacking approaches	
Neural net	6433.092
OLS	6463.73
Lasso with CV (TWI)	6780.161
Ridge with CV (TWI)	6760.134
Lasso with CV (QSI)	5722.624
Ridge with CV (QSI)	5995.346
Random forest (low regularization)	6089.389
Random forest (high regularization)	6552.221
Gradient boosting (low regularization)	7003.373
Gradient boosting (high regularization)	7992.538

Notes: In the case of DDML estimators, the average estimates and standard errors are based on 50 replications. Panel A is reproduced from Table 1 in WZ.

Table B.2: Short-stacking weights using CLS

Estimator	*Observations*						
	200	400	600	800	1 200	1 600	9 915
Panel A. $E[Y\|X]$, $K = 10$							
OLS	.164	.152	.115	.079	.037	.019	0
Neural net	.047	.045	.048	.067	.098	.05	.076
Lasso with CV (TWI)	.043	.034	.034	.035	.03	.033	.091
Ridge with CV (TWI)	.056	.048	.041	.025	.011	.006	.032
Lasso with CV (QSI)	.252	.274	.266	.264	.271	.297	.639
Ridge with CV (QSI)	.194	.252	.297	.328	.341	.357	.153
Random forest (low regularization)	.095	.097	.113	.131	.161	.2	.01
Random forest (high regularization)	.081	.04	.025	.021	.018	.016	0
Gradient boosting (low regularization)	.041	.04	.049	.041	.03	.021	0
Gradient boosting (high regularization)	.028	.019	.013	.009	.002	.001	0
Panel B. $E[D\|X]$, $K = 10$							
OLS	.132	.196	.234	.252	.245	.257	.163
Neural net	.04	.041	.038	.036	.031	.029	.038
Lasso with CV (TWI)	.053	.031	.025	.02	.016	.012	.106
Ridge with CV (TWI)	.038	.018	.013	.015	.008	.005	.029
Lasso with CV (QSI)	.173	.225	.25	.248	.25	.228	.413
Ridge with CV (QSI)	.202	.124	.072	.06	.068	.064	0
Random forest (low regularization)	.103	.123	.144	.187	.249	.307	.006
Random forest (high regularization)	.159	.129	.107	.09	.051	.031	.102
Gradient boosting (low regularization)	.043	.046	.054	.047	.045	.041	.144
Gradient boosting (high regularization)	.059	.065	.064	.046	.038	.025	0
Panel C. $E[Y\|X]$, $K = 10$							
OLS	.122	.098	.066	.026	.003	.001	0
Neural net	0	0	0	0	0	0	0
Lasso with CV (TWI)	.03	.022	.01	.013	.014	.023	0
Ridge with CV (TWI)	.074	.077	.079	.052	.03	.013	0
Lasso with CV (QSI)	.323	.376	.361	.381	.393	.405	.995
Ridge with CV (QSI)	.239	.314	.379	.428	.478	.479	.005
Random forest (low regularization)	.129	.058	.05	.049	.049	.044	0
Random forest (high regularization)	.022	.005	.001	.001	0	.001	0
Gradient boosting (low regularization)	.025	.033	.046	.046	.032	.034	0
Gradient boosting (high regularization)	.035	.016	.009	.004	0	0	0
Panel D. $E[D\|X]$, $K = 10$							
OLS	.038	.108	.17	.189	.173	.132	.005
Neural net	0	0	0	0	0	0	0
Lasso with CV (TWI)	.058	.032	.017	.011	.005	.003	.002
Ridge with CV (TWI)	.06	.013	.009	.01	.002	.001	0
Lasso with CV (QSI)	.232	.309	.313	.287	.261	.168	.754
Ridge with CV (QSI)	.242	.105	.032	.034	.05	.032	0
Random forest (low regularization)	.079	.028	.011	.008	.004	.003	0
Random forest (high regularization)	.185	.249	.256	.304	.344	.507	0
Gradient boosting (low regularization)	.009	.022	.048	.064	.115	.141	.24
Gradient boosting (high regularization)	.098	.135	.143	.092	.046	.013	0

Notes: The table reports the stacking weights corresponding to the DDML stacking estimator in Figure 3. The stacking weights are averaged over folds, based on 10-fold cross-fitting and shows for the estimation of $E[Y|X]$ and $E[D|X]$ in Panel A and B, respectively. See notes below Table 3 for more information.

Table B.3: Mean bias under linear DGP in small samples based on the calibrated Monte Carlo in Section 4.1

	Bootstrap sample size n_b									
	200		400		800		1 600		9 915	
	Bias	s.e.	Bias	s.e.	Bias	s.e.	Bias	s.e.	Bias	s.e.
Full sample estimators:										
OLS	230.4	272.9	-28.4	198.8	9.7	140.9	11.7	102.0	78.3	39.1
PDS-Lasso	-714.3	270.5	-739.5	197.3	-367.9	141.1	-76.3	102.3	78.8	39.1
DDML methods:										
Candidate learners (K = 10)										
OLS	179.4	275.7	-85.1	199.2	2.5	141.0	7.9	102.0	77.8	39.1
Lasso with CV (2nd order poly)	357.0	271.3	-38.9	199.6	3.4	141.7	9.1	102.1	77.5	39.1
Ridge with CV (2nd order poly)	334.6	267.0	6.4	197.5	10.8	142.0	-9.9	102.0	77.5	39.2
Lasso with CV (10th order poly)	-4864.6	5369.1	754.7	957.7	-476.8	1663.2	-547.1	454.5	141.0	46.4
Ridge with CV (10th order poly)	-14274.3	10353.7	1038.4	7241.9	1596.6	3409.3	1442.5	1128.7	449.8	62.2
Random forest (low regularization)	328.1	302.1	-112.2	220.3	-148.4	153.2	-69.6	112.1	-41.8	45.7
Random forest (high regularization)	1086.7	274.9	255.6	201.6	206.3	143.1	128.5	103.7	68.4	39.9
Gradient boosting (low regularization)	171.1	300.8	-280.4	210.3	-149.1	145.7	-60.3	104.2	44.3	39.3
Gradient boosting (high regularization)	389.4	286.4	-77.4	203.4	65.0	142.9	75.2	102.6	139.1	39.2
Neural net	3887.7	247.4	3926.3	199.2	3504.2	143.1	1874.9	104.9	212.1	39.2
Meta learners (K = 10)										
Stacking: CLS	-1150.3	1159.2	-84.3	224.9	28.2	165.0	47.1	111.1	73.1	39.2
Stacking: Average	-3471.2	3073.6	195.7	358.6	-71.7	357.8	368.6	204.4	86.3	39.3
Stacking: OLS	4.2e6	5.4e6	-1040.5	2688.0	-2085.4	1149.5	-365.2	255.9	32.8	55.8
Stacking: Single-best	-5.0	275.9	-212.2	211.7	-57.8	141.7	-27.6	102.4	68.2	39.1
Short-stacking: CLS	322.3	271.8	-54.8	199.6	51.6	141.2	25.1	102.1	73.2	39.1
Short-stacking: Average	-3471.2	3073.6	195.7	358.6	-71.7	357.8	368.6	204.4	86.3	39.3
Short-stacking: OLS	-214.9	269.4	-278.5	200.0	-100.4	140.4	-41.0	101.6	70.7	39.1
Short-stacking: Single-best	189.1	271.1	-91.4	199.1	0.7	140.8	3.1	101.9	75.9	39.1
Pooled stacking: CLS	465.6	271.7	39.7	199.8	89.6	140.9	18.0	102.1	73.7	39.1
Pooled stacking: Average	-3471.2	3073.6	195.7	358.6	-71.7	357.8	368.6	204.4	86.3	39.3
Pooled stacking: OLS	326.6	272.8	-97.4	200.0	-68.3	141.3	-23.2	102.3	72.6	39.1
Pooled stacking: Single-best	150.8	270.7	-93.1	199.3	-11.6	141.2	8.3	102.1	76.6	39.1
Meta learners (K = 2)										
Stacking: CLS	6329.4	5587.9	-66.9	531.5	-21.5	200.4	261.0	111.9	42.8	40.2
Stacking: Average	-972.6	2751.4	1796.8	1765.4	254.6	647.7	815.2	215.7	107.7	51.2
Stacking: OLS	-10699.3	13002.6	3.8e7	3.8e7	-63.3	2752.6	-1043.6	547.8	-129.3	96.2
Stacking: Single-best	5893.9	5319.3	73.8	249.9	-158.5	149.5	115.1	99.1	43.7	39.9
Short-stacking: CLS	389.3	285.0	314.6	210.6	9.9	144.1	218.4	98.2	45.0	39.7
Short-stacking: Average	-972.6	2751.4	1796.8	1765.4	254.6	647.7	815.2	215.7	107.7	51.2
Short-stacking: OLS	-207.9	275.1	84.7	206.1	-113.2	141.6	124.9	97.9	37.6	39.6
Short-stacking: Single-best	241.8	284.2	209.2	210.2	-37.5	145.1	178.8	98.5	44.4	39.7
Pooled stacking: CLS	-22063.1	22632.2	378.9	322.2	-141.5	166.8	230.4	107.1	58.6	41.4
Pooled stacking: Average	-972.6	2751.4	1796.8	1765.4	254.6	647.7	815.2	215.7	107.7	51.2
Pooled stacking: OLS	-26101.3	29649.6	-2324.4	2415.4	-1641.0	1777.9	-79.0	272.3	46.5	47.8
Pooled stacking: Single-best	12247.4	12403.7	122.1	214.6	-88.2	146.2	156.7	98.6	46.9	39.7

Notes: The table reports mean bias and associated standard errors for the listed estimators. We consider DDML with the following individual 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, maximum depth of 3 and a learning rate of 0.01), gradient-boosted trees with high regularization: 250 trees, maximum depth of 3 and a learning 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.

Table B.4: Mean bias under non-linear DGP in small samples based on the calibrated Monte Carlo in Section 4.1

	Bootstrap sample size n_b									
	200		400		800		1 600		9 915	
	Bias	s.e.	Bias	s.e.	Bias	s.e.	Bias	s.e.	Bias	s.e.
Full sample estimators:										
OLS	-2339.8	331.6	-2208.9	226.7	-2332.7	165.1	-2440.2	112.5	-2586.2	47.6
PDS-Lasso	-2281.9	329.6	-2740.9	224.9	-2789.1	168.4	-2716.2	114.1	-2597.3	47.7
DDML methods:										
Candidate learners ($K = 10$)										
OLS	-2864.1	370.5	-2607.8	247.4	-2479.6	172.1	-2526.7	115.0	-2606.5	47.8
Lasso with CV (2nd order poly)	-1048.1	357.5	-730.3	232.5	243.2	166.1	467.6	111.8	783.4	43.0
Ridge with CV (2nd order poly)	-1730.2	348.3	-1127.8	232.0	481.5	170.6	833.2	113.3	806.7	43.0
Lasso with CV (10th order poly)	22248.8	25988.5	-2907.1	2774.5	1598.1	3131.5	4485.9	3456.2	185.6	457.2
Ridge with CV (10th order poly)	24105.0	16649.8	2351.6	2060.6	-5341.5	5329.5	-602.5	2313.4	2873.4	552.4
Random forest (low regularization)	-84.4	343.4	18.2	231.6	-5.4	158.3	-89.8	113.2	-63.6	47.7
Random forest (high regularization)	-159.0	333.3	-37.4	223.1	99.8	154.6	-77.1	107.6	-78.1	42.7
Gradient boosting (low regularization)	-281.5	351.7	-32.4	238.7	-11.1	157.8	34.1	108.8	64.6	41.3
Gradient boosting (high regularization)	-89.7	336.4	208.6	229.0	179.2	152.6	182.9	105.2	218.2	41.2
Neural net	1839.2	299.8	2045.7	225.2	1735.2	159.0	209.6	107.6	-472.8	43.6
Meta learners ($K = 10$)										
Stacking: CLS	11936.9	14230.0	-1198.8	965.2	803.6	583.6	915.4	1141.7	437.8	46.8
Stacking: Average	7161.2	5979.0	498.4	566.1	-368.8	751.2	765.1	482.6	451.8	50.7
Stacking: OLS	-2.7e7	2.4e7	-3391.9	2466.9	3033.6	3386.2	6271.2	4906.4	608.4	190.2
Stacking: Single-best	3472.5	4282.0	-385.1	306.5	687.0	541.4	542.5	255.2	154.1	46.1
Short-stacking: CLS	-278.6	311.4	-117.9	208.7	88.3	146.0	188.6	101.9	157.7	41.0
Short-stacking: Average	7161.2	5979.0	498.4	566.1	-368.8	751.2	765.1	482.6	451.8	50.7
Short-stacking: OLS	-524.0	298.3	-209.4	199.9	16.5	142.2	88.7	100.2	126.1	40.8
Short-stacking: Single-best	-602.8	310.3	-363.9	215.6	51.5	150.3	196.2	104.5	66.9	41.3
Pooled stacking: CLS	-382.2	315.4	-319.7	218.0	-73.3	150.7	143.9	103.7	179.9	41.0
Pooled stacking: Average	7161.2	5979.0	498.4	566.1	-368.8	751.2	765.1	482.6	451.8	50.7
Pooled stacking: OLS	-217.6	359.4	-231.1	223.5	-223.3	158.8	-23.4	120.8	152.0	40.9
Pooled stacking: Single-best	-679.0	319.8	-316.5	228.0	-52.0	154.5	159.4	105.9	68.2	41.2
Meta learners ($K = 2$)										
Stacking: CLS	656.2	1786.0	583.5	718.1	-393.0	500.7	-710.3	585.3	133.9	195.6
Stacking: Average	2921.9	1220.8	2639.9	1126.1	318.6	1095.6	-656.9	526.5	94.0	110.3
Stacking: OLS	2.2e7	1.9e7	-8671.2	9731.4	-5593.2	7804.4	971.5	3368.2	-204.8	405.2
Stacking: Single-best	2892.1	1501.3	-442.5	347.7	-387.1	295.4	-311.5	506.8	-121.9	272.9
Short-stacking: CLS	410.7	316.1	-76.2	215.5	-320.8	163.2	-219.0	107.8	162.7	42.0
Short-stacking: Average	2921.9	1220.8	2639.9	1126.1	318.6	1095.6	-656.9	526.5	94.0	110.3
Short-stacking: OLS	-130.7	287.7	-201.9	202.5	-322.2	155.3	-226.9	106.1	123.6	41.7
Short-stacking: Single-best	-28.4	323.0	-363.6	221.3	-546.6	164.0	-320.3	110.4	71.7	42.6
Pooled stacking: CLS	767.3	1131.7	942.5	577.0	-673.3	318.3	-734.7	567.8	209.8	63.7
Pooled stacking: Average	2921.9	1220.8	2639.9	1126.1	318.6	1095.6	-656.9	526.5	94.0	110.3
Pooled stacking: OLS	-1784.7	5455.8	-6585.3	4193.1	-897.4	1388.9	903.3	2480.1	234.5	124.9
Pooled stacking: Single-best	1594.5	1414.5	-26.1	449.7	-557.8	366.5	-176.4	379.9	103.3	45.8

Notes: See Table B.3 notes.

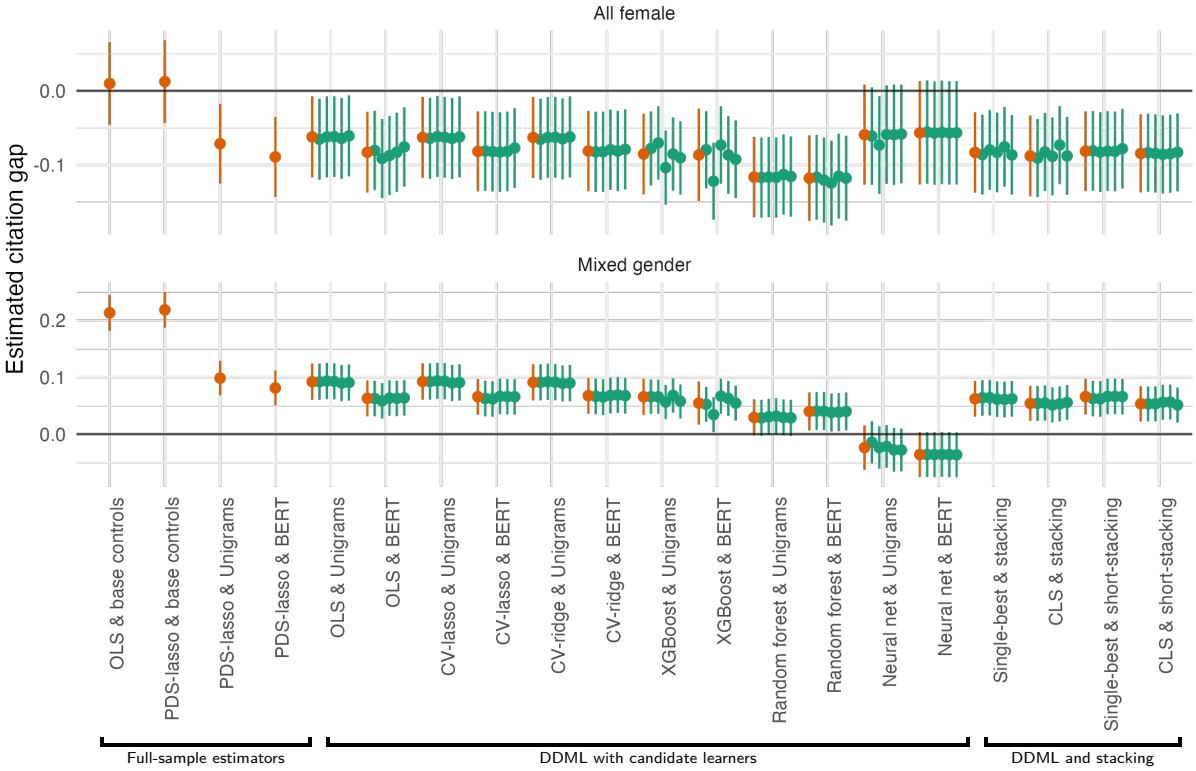
Table B.5: Coverage in small samples based on the calibrated Monte Carlo in Section 4.1

	Bootstrap sample size n_b									
	Panel A. Linear DGP					*Panel B. Non-linear DGP*				
	200	400	800	1600	9915	200	400	800	1 600	9 915
Full sample estimators:										
OLS	0.95	0.95	0.95	0.94	0.95	0.92	0.95	0.94	0.92	0.59
PDS-Lasso	0.96	0.95	0.94	0.94	0.95	0.92	0.95	0.93	0.91	0.59
DDML methods:										
Candidate learners ($K = 10$)										
OLS	0.95	0.95	0.94	0.94	0.94	0.93	0.95	0.94	0.91	0.59
Lasso with CV (2nd order poly)	0.95	0.95	0.94	0.93	0.94	0.94	0.95	0.96	0.94	0.90
Ridge with CV (2nd order poly)	0.96	0.95	0.94	0.94	0.94	0.94	0.96	0.96	0.94	0.90
Lasso with CV (10th order poly)	0.90	0.92	0.91	0.93	0.94	0.87	0.85	0.86	0.89	0.95
Ridge with CV (10th order poly)	0.82	0.87	0.88	0.88	0.95	0.78	0.83	0.84	0.87	0.95
Random forest (low regularization)	0.92	0.93	0.92	0.91	0.91	0.92	0.93	0.94	0.93	0.90
Random forest (high regularization)	0.95	0.94	0.95	0.94	0.95	0.94	0.95	0.96	0.95	0.94
Gradient boosting (low regularization)	0.92	0.93	0.93	0.94	0.95	0.92	0.93	0.95	0.95	0.95
Gradient boosting (high regularization)	0.94	0.95	0.94	0.94	0.95	0.93	0.94	0.97	0.95	0.94
Neural net	0.94	0.92	0.90	0.90	0.95	0.94	0.95	0.95	0.97	0.94
Meta learners ($K = 10$)										
Stacking: CLS	0.94	0.95	0.94	0.93	0.95	0.92	0.94	0.94	0.94	0.95
Stacking: Average	0.94	0.95	0.94	0.94	0.95	0.90	0.93	0.95	0.95	0.95
Stacking: OLS	0.85	0.89	0.89	0.90	0.93	0.82	0.86	0.89	0.92	0.93
Stacking: Single-best	0.94	0.94	0.95	0.94	0.95	0.94	0.94	0.95	0.95	0.94
Short-stacking: CLS	0.95	0.95	0.94	0.94	0.94	0.94	0.96	0.96	0.95	0.94
Short-stacking: Average	0.94	0.95	0.94	0.94	0.95	0.90	0.93	0.95	0.95	0.95
Short-stacking: OLS	0.95	0.94	0.95	0.94	0.95	0.94	0.95	0.95	0.95	0.94
Short-stacking: Single-best	0.95	0.95	0.94	0.94	0.94	0.93	0.96	0.96	0.96	0.95
Pooled stacking: CLS	0.95	0.95	0.94	0.94	0.94	0.94	0.95	0.96	0.95	0.94
Pooled stacking: Average	0.94	0.95	0.94	0.94	0.95	0.90	0.93	0.95	0.95	0.95
Pooled stacking: OLS	0.95	0.95	0.94	0.94	0.94	0.94	0.96	0.96	0.95	0.94
Pooled stacking: Single-best	0.95	0.95	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95
Meta learners ($K = 2$)										
Stacking: CLS	0.90	0.90	0.93	0.94	0.94	0.90	0.92	0.93	0.93	0.94
Stacking: Single-best	0.91	0.91	0.93	0.94	0.94	0.92	0.93	0.93	0.93	0.94
Short-stacking: CLS	0.94	0.93	0.94	0.95	0.94	0.93	0.94	0.93	0.94	0.94
Short-stacking: Average	0.90	0.91	0.93	0.93	0.94	0.89	0.90	0.91	0.92	0.94
Short-stacking: OLS	0.95	0.94	0.94	0.94	0.94	0.94	0.94	0.93	0.95	0.94
Short-stacking: Single-best	0.94	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.95	0.94
Pooled stacking: CLS	0.93	0.90	0.93	0.94	0.95	0.92	0.94	0.93	0.95	0.94
Pooled stacking: Average	0.90	0.91	0.93	0.93	0.94	0.89	0.90	0.91	0.92	0.94
Pooled stacking: OLS	0.88	0.89	0.92	0.93	0.94	0.85	0.90	0.90	0.93	0.94
Pooled stacking: Single-best	0.94	0.92	0.93	0.95	0.94	0.94	0.95	0.94	0.95	0.94

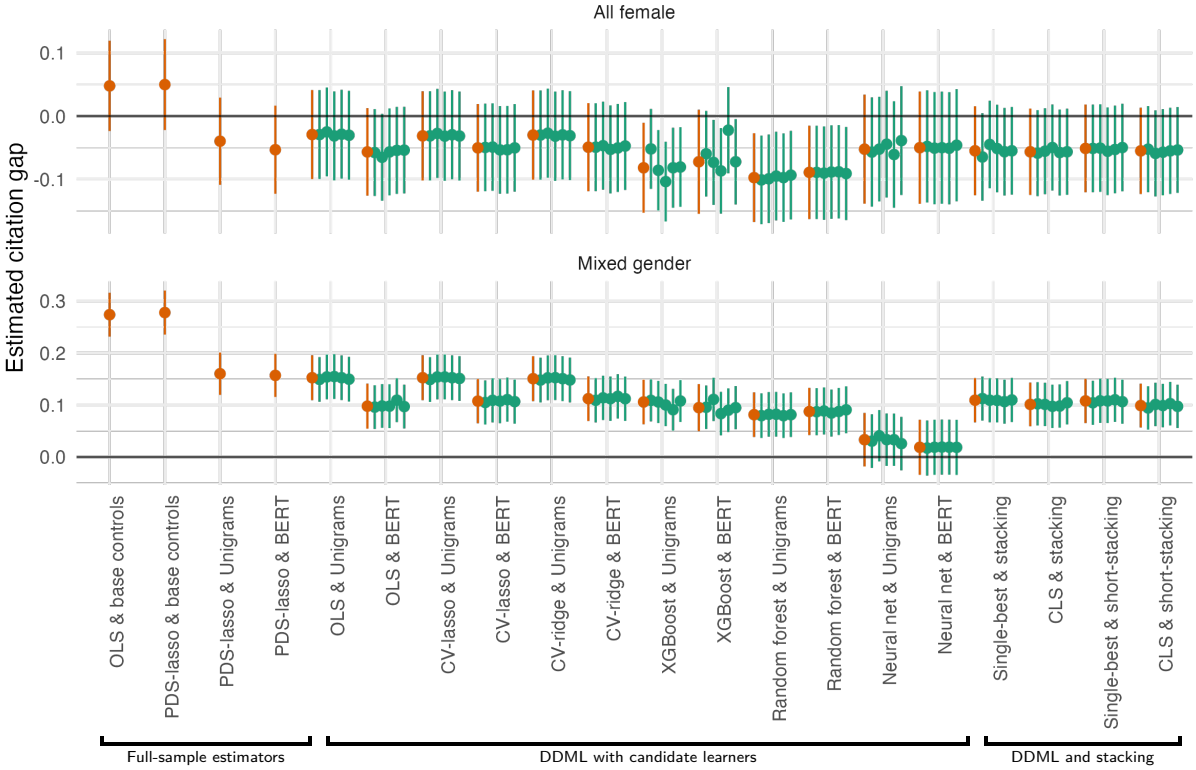
Notes: This table reports coverage of 95% interval estimates in the small sample simulation. See Table B.3 notes for more detail.

C Gender citation gap

Figure C.1: The citation gap by authors' gender composition



(a) Threshold 60%



(b) Threshold 90%

Notes: The figure shows estimates of θ_0 summarizing average relative difference in total citations between all-male and all-female authorship, and all-male and mixed-gender authorship, respectively using different thresholds for successful classification of an author's sex. See Figure 6 notes for more information.

Table C.1: Estimates for the citation penalty of all-female and mixed-gender authored articles

	Log citations		Citation counts	
	All female	*Mixed gender*	*All female*	*Mixed gender*
Panel A. Full-sample estimators				
OLS & base controls	0.022	0.226***	-9.129**	12.168***
	(0.03)	(0.017)	(4.438)	(3.045)
PDS-lasso & base controls	0.025	0.231***	-8.901**	12.317***
	(0.03)	(0.017)	(4.431)	(3.045)
PDS-lasso & Unigrams	-0.056*	0.106***	-15.77***	2.844
	(0.029)	(0.017)	(4.283)	(3.053)
Panel B. DDML with candidate learners				
PDS-lasso & BERT	-0.074**	0.092***	-15.051***	2.125
	(0.029)	(0.017)	(4.428)	(3.053)
OLS & Unigrams	-0.052*	0.103***	-12.293**	2.647
	(0.03)	(0.018)	(6.104)	(3.635)
OLS & BERT	-0.072**	0.066***	-8.686	1.454
	(0.029)	(0.017)	(6.242)	(3.655)
CV-lasso & Unigrams	-0.05*	0.104***	-12.388**	2.648
	(0.03)	(0.018)	(6.116)	(3.621)
CV-lasso & BERT	-0.069**	0.069***	-9.928	2.209
	(0.029)	(0.017)	(6.142)	(3.648)
CV-ridge & Unigrams	-0.051*	0.102***	-12.383**	2.077
	(0.03)	(0.018)	(6.115)	(3.627)
CV-ridge & BERT	-0.067**	0.075***	-9.86	2.056
	(0.029)	(0.017)	(6.155)	(3.636)
XGBoost & Unigrams	-0.065**	0.073***	37.984***	23.481***
	(0.03)	(0.02)	(6.655)	(3.629)
XGBoost & BERT	-0.07**	0.073***	9.473	8.922**
	(0.031)	(0.021)	(7.306)	(3.796)
Random forest & Unigrams	-0.103***	0.044**	0.444	5.255
	(0.029)	(0.017)	(6.098)	(3.623)
Random forest & BERT	-0.106***	0.055***	8.919	10.869***
	(0.031)	(0.018)	(6.148)	(3.667)
Neural net & Unigrams	-0.05	0.005	-16.867***	-9.791***
	(0.036)	(0.021)	(6.317)	(3.647)
Neural net & BERT	-0.047	-0.014	-16.979***	-10.31***
	(0.038)	(0.022)	(6.316)	(3.643)
Panel C. DDML with stacking approaches				
Single-best & stacking	-0.07**	0.072***	-10.204*	2.274
	(0.029)	(0.017)	(6.123)	(3.65)
CLS & stacking	-0.071**	0.063***	-9.485	1.971
	(0.029)	(0.017)	(6.103)	(3.635)
Single-best & short-stacking	-0.069**	0.07***	-9.916	2.254
	(0.029)	(0.017)	(6.143)	(3.643)
CLS & short-stacking	-0.07**	0.062***	-9.529	1.615
	(0.029)	(0.017)	(6.099)	(3.628)

Notes: The table shows median-aggregated estimates of the gender citation gap for all–female and mixed-gender authored articles. We show results using both log citations and citation counts as the outcome variable. Standard errors are robust to heteroskedasticity. See Figure 6 for information on the candidate learners and stacking approaches.

D Gender wage gap

Table D.1: Stacking weights in the gender wage gap application.

	Conventional stacking			*Short-stacking*			*Mean-squared error*		
	$g_0(0,X)$	$g_0(1,X)$	$m_0(X)$	$g_0(0,X)$	$g_0(1,X)$	$m_0(X)$	$g_0(0,X)$	$g_0(1,X)$	$m_0(X)$
OLS/logit	0.023	0.012	0.242	0.027	0.013	0.211	0.369	0.347	0.161
OLS/logit (simple)	0.004	0.	0.	0.	0.	0.	0.267	0.204	0.223
CV-lasso	0.103	0.136	0.109	0.03	0.076	0.047	0.236	0.178	0.16
CV-ridge	0.189	0.04	0.064	0.225	0.024	0.108	0.237	0.18	0.161
CV-lasso (extended)	0.041	0.157	0.016	0.035	0.266	0.002	0.238	0.18	0.161
CV-ridge (extended)	0.011	0.04	0.011	0.003	0.024	0.022	0.336	0.194	0.161
Random forest 1	0.435	0.506	0.275	0.483	0.507	0.28	0.23	0.176	0.161
Random forest 2	0.	0.	0.	0.	0.	0.	0.258	0.19	0.171
Random forest 3	0.	0.	0.	0.	0.	0.	0.274	0.199	0.179
Gradient boosting 1	0.025	0.008	0.039	0.011	0.003	0.022	0.239	0.183	0.16
Gradient boosting 2	0.15	0.059	0.216	0.175	0.063	0.285	0.254	0.196	0.161
Neural net 1	0.013	0.022	0.	0.	0.	0.	0.349	0.263	0.241
Neural net 2	0.008	0.02	0.027	0.01	0.023	0.023	0.643	0.357	0.176

Notes: The table shows weights of conventional and short-stacking along with the mean-squared prediction error by candidate learners and by variable. The final learner is constrained least squares. The stacking weights are averaged over cross-fitting repetitions. Pooled stacking weights are shown in Appendix Table D.2.

Table D.2: Stacking weights of pooled stacking using constrained least squares.

	Pooled stacking		
	$g_0(0,X)$	$g_0(1,X)$	$m_0(X)$
OLS/logit	0.014	0.001	0.257
OLS/logit (simple)	0.	0.	0.
CV-lasso	0.15	0.23	0.136
CV-ridge	0.205	0.064	0.063
CV-lasso (extended)	0.	0.078	0.
CV-ridge (extended)	0.	0.019	0.
Random forest 1	0.462	0.521	0.288
Random forest 2	0.	0.	0.
Random forest 3	0.	0.	0.
Gradient boosting 1	0.	0.	0.008
Gradient boosting 2	0.165	0.071	0.23
Neural net 1	0.	0.	0.
Neural net 2	0.004	0.016	0.018

Notes: The table shows pooled stacking weights for each of the considered candidate learners. The final learner is constrained least squares. The stacking weights are averaged over cross-fitting repetitions.

Table D.3: Stacking weights using single-best final learner.

	Conventional stacking			*Short-stacking*			*Pooled stacking*		
	$g_0(0,X)$	$g_0(1,X)$	$m_0(X)$	$g_0(0,X)$	$g_0(1,X)$	$m_0(X)$	$g_0(0,X)$	$g_0(1,X)$	$m_0(X)$
OLS/logit	0.	0.	0.	0.	0.	0.	0.	0.	0.
OLS/logit (simple)	0.	0.	0.	0.	0.	0.	0.	0.	0.
CV-lasso	0.06	0.03	0.79	0.	0.	1.	0.	0.	0.4
CV-ridge	0.07	0.	0.04	0.	0.	0.	0.	0.	0.1
CV-lasso (extended)	0.02	0.06	0.03	0.	0.	0.	0.	0.	0.1
CV-ridge (extended)	0.	0.	0.01	0.	0.	0.	0.	0.	0.
Random forest 1	0.85	0.91	0.02	1.	1.	0.	1.	1.	0.
Random forest 2	0.	0.	0.	0.	0.	0.	0.	0.	0.
Random forest 3	0.	0.	0.	0.	0.	0.	0.	0.	0.
Gradient boosting 1	0.	0.	0.11	0.	0.	0.	0.	0.	0.4
Gradient boosting 2	0.	0.	0.	0.	0.	0.	0.	0.	0.
Neural net 1	0.	0.	0.	0.	0.	0.	0.	0.	0.
Neural net 2	0.	0.	0.	0.	0.	0.	0.	0.	0.

Notes: The table shows weights of conventional stacking, short-stacking and pooled stacking by candidate learners and by conditional expectation function. The stacking weights are averaged over cross-fitting repetitions.

Table D.4: Median aggregate estimates by stacking approach and by final learner

	Final learner			
	Unweighted average	*CLS*	*OLS*	*Single-best*
Regular stacking	-0.101	-0.075	-0.197	-0.061
	(0.017)*	(0.028)*	(11.894)	(0.069)
Short- stacking	-0.101	-0.076	-0.001	-0.085
	(0.017)*	(0.028)*	(0.184)	(0.065)

Notes: The table reports median aggregate estimates by stacking type and final learner. See Figure 7 for more information.

Table D.5: Median aggregate estimates for each candidate learner

	Gender wage gap
OLS/logit	-0.12
	(0.094)
CV-lasso	-0.067
	(0.063)
CV-ridge	-0.064
	(0.09)
OLS/logit (simple)	-0.12
	(0.016)*
CV-lasso (extended)	-0.055
	(0.076)
CV-ridge (extended)	-0.173
	(0.19)
Random forest 1	-0.079
	(0.023)*
Random forest 2	-0.105
	(0.016)*
Random forest 3	-0.11
	(0.015)*
Gradient boosting 1	-0.075
Observations	4836

Notes: The table reports median aggregate estimates by candidate learner. See Figure 7 for more information.

Check for updates

Received: 8 June 2023 | Revised: 18 July 2024 | Accepted: 18 July 2024

DOI: 10.1002/jae.3092

WILEY Journal of APPLIED ECONOMETRICS

RESEARCH ARTICLE

Optimal multi-action treatment allocation: A two-phase field experiment to boost immigrant naturalization

Achim Ahrens | **Alessandra Stampi-Bombelli** | **Selina Kurer** | **Dominik Hangartner**

Immigration Policy Lab, ETH Zurich, Zurich, Switzerland

Correspondence
Achim Ahrens, Immigration Policy Lab, ETH Zurich, Zurich, Switzerland.
Email: achim.ahrens@gess.ethz.ch

Funding information
This research was supported by the Stiftung Mercator Schweiz and by the *NCCR - on the move* program, which is funded by the Swiss National Science Foundation (grant no. 51NF40-182897).

Summary

Research underscores the role of naturalization in enhancing immigrants' socio-economic integration, yet application rates remain low. We estimate a policy rule for a letter-based information campaign encouraging newly eligible immigrants in Zurich, Switzerland, to naturalize. The policy rule assigns one out of three treatment letters to each individual, based on their observed characteristics. We field the policy rule to one-half of 1717 immigrants, while sending random treatment letters to the other half. Despite only moderate treatment effect heterogeneity, the policy tree yields a larger, albeit insignificant, increase in application rates compared with assigning the same letter to everyone.

KEYWORDS

immigrant naturalization, policy learning, randomized field experiment, statistical decision rules, targeted treatment

1 | INTRODUCTION

Policymakers frequently need to select among alternative treatment options. While one of the stated aims of empirical research is to provide new insights to inform decision-making processes, the primary focus is usually on estimating averages of treatment effects rather than providing direct guidance on how to design assignment mechanisms for alternative treatments. In practice, the empirical researcher specifies a statistical model and estimates the efficacy of each treatment using an experimental or observational sample, while the decision-maker assigns the treatment, interpreting the point estimates as if they were true. This approach, termed *as-if* maximization by Manski (2021), tends to yield one-size-fits-all rules assigning the same treatment to the wider population. Such one-size-fits-all policies seem inefficient given that treatment effects frequently exhibit relevant effect heterogeneity across observations and the increasing availability of administrative data providing rich individual characteristics.

Policy learning provides a framework for directly estimating statistical decision rules, so-called policy rules, which prescribe treatments to individuals based on their observed characteristics (also known as profiling or targeting). While its origins date back to statistical decision theory (Savage, 1951; Wald, 1950), the seminal work of Manski (2004) sparked a flourishing literature in econometrics which has developed methods for estimating statistical treatment rules, initially focusing on data drawn from randomized control trials (Hirano & Porter, 2009; Manski, 2004; Stoye, 2009; 2012) but subsequently also covering observational data under unconfoundedness assumptions (Athey & Wager, 2021; Manski, 2007; Zhou et al., 2022; see Hirano & Porter, 2020 for a review). While applied research using policy learning is still relatively scarce, previous work has revealed the potential for data-driven treatment allocation across a variety of domains, including active labor market programs (e.g., Frölich, 2008; Lechner & Smith, 2007), vaccines accounting for spill-over effects

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2024 The Author(s). Journal of Applied Econometrics published by John Wiley & Sons, Ltd.

(Kitagawa & Wang, 2023), deforestation-reducing policies (Assunção et al., 2022), antimalaria subsidies under budget constraints (Bhattacharya & Dupas, 2012), energy use information campaigns (Ida et al., 2022; Gerarden & Yang, 2022), and maximizing fundraising (Cagala et al., 2021).

In this preregistered study, we co-design and evaluate an individualized treatment allocation program with the goal of facilitating the naturalization of eligible immigrants in the City of Zurich, Switzerland. An expanding body of literature is utilizing close naturalization referendums or temporal discontinuities created by policy reform to enable credible comparisons between naturalized and nonnaturalized immigrants to demonstrate that acquiring host-country citizenship offers long-term integration benefits for immigrants and, indirectly, to host societies. These benefits span various integration dimensions, including employment and earnings (Gathmann & Keller, 2018; Hainmueller et al., 2019), political efficacy, knowledge, and participation (Hainmueller et al., 2015), as well as social incorporation and cooperation (Felfe et al., 2021; Hainmueller et al., 2017; Keller et al., 2015). Yet, despite these benefits, naturalization rates remain low in many countries, with a median annual naturalization rate (number of naturalized immigrants divided by number of immigrants) of 1.9% in Europe and 3.1% in the U.S. (Ward et al., 2019). Against this background, policymakers at the national level in Estonia, Ireland, Latvia, North Macedonia, Spain and at the local level in Germany, Italy, Switzerland, and the USA have deployed information campaigns to boost citizenship applications and fully reap the integration benefits of naturalization (Huddleston, 2013).

Informed by existing research (e.g., Bauböck et al., 2006; Bloemraad, 2002; National Academies of Sciences, et al., 2016) and the insights of integration and naturalization bureaucrats of the City of Zurich, this study considers interventions that address three specific hurdles blocking eligible immigrants' path to citizenship. These hurdles include the following: (i) the perceived complexity of the application process, (ii) knowledge gaps about the requirements for naturalization, and (iii) the feeling of not being welcome to naturalize. To address the first two hurdles, we co-designed specific information letters with the City of Zurich. For the third hurdle, a letter sent by the Mayor of City of Zurich encouraged immigrants to apply. In line with recent recommendations by Haaland et al. (2023), we opted for three separate treatment letters with accompanying flyers to ensure that each letter is short and easy to understand. Addressing all hurdles in a combined treatment letter with several flyers is likely counterproductive due to the limited time and attention that recipients devote when reading the letters.[2]

Since it is unknown which treatment letter is optimal for maximizing the individual application probabilities and given that the optimal treatment choice may differ among individuals, we derive a multi-action policy rule. This policy rule is structured as a decision tree, which is referred to as a "policy tree." Policy trees are introduced by Athey and Wager (2021) for binary and by Zhou et al. (2022) for multivalued treatments. In our context, the policy tree selects one treatment from a set of three treatment options for each eligible immigrant based on their individual characteristics including residency, nationality, and age. The treatment options are incorporated into three different letters with enclosed flyers sent out by the City of Zurich. Thus, by applying policy learning, we allow the optimal content and framing of the information provision to vary with observed immigrant characteristics. The policy rule is chosen to maximize the application rate for naturalization, the first step in the process of acquiring Swiss citizenship.

Policy trees possess several strengths that make them a particularly promising method for immigrant naturalization and other sensitive policy contexts. First, policy trees allow policymakers and researchers to select those variables that can be used to tailor treatment assignment and, more importantly, exclude those that should not be used (e.g., protected characteristics such as religion) and quantify the costs of exclusion in terms of foregone treatment efficacy. Second, policy trees make transparent which variables, and which variable values, guide treatment assignment. This is in contrast to black-box *plug-in* rules, providing no insights into what drives treatment allocation. Related to the second strength is the third: Policy trees are easy to visualize and easy to explain to users of the research—for example, policymakers, case officers, and subjects receiving treatment assignment—even if they lack training in statistics. Together, transparency and interpretability are important steps towards satisfying requirements for explainable artificial intelligence (AI), for example, as outlined in recent proposals for the regulation of AI by the European Commission (2021) and The White House (2022). Finally, from a practical perspective, the so-called offline approach of policy trees, which learns policies from a single data batch, is often easier to implement in a public policy context than adaptive approaches training policy rules dynamically over time (e.g., Caria et al., 2020).

After introducing the methodology of policy learning, we illustrate the practical feasibility of the targeted assignment rule and evaluate its benefits using a tailored, two-phase randomized controlled trial. In the first phase of our field exper-

[2]A large literature in behavioral economics stresses that information processing is costly and provides evidence that individuals often fail to translate all available information into optimal decisions (for recent reviews, see Handel & Schwartzstein, 2018; Gabaix, 2019; Maćkowiak et al., 2023).

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

iment, we randomly allocate 60% of our sample of 5145 citizenship-eligible immigrants to receive one of three letters addressing specific naturalization hurdles. Based on first-wave application outcomes and leveraging observed treatment effect heterogeneity, we estimate the optimal multi-action policy rule using the estimation framework of Zhou et al. (2022). In the second phase, we field the fitted policy rule on one-half of the remaining sample while sending random treatment letters to the other half. Adopting terminology from reinforcement learning, we refer to these two phases as the exploration phase (aimed at gathering knowledge about treatment efficacy) and the exploitation phase (aimed at implementing the reward-maximizing strategy), respectively. We evaluate the performance of the derived policy rule against random treatment allocation, one-size-fits-all policy rules assigning the same treatment to everyone, and a model-free *plug-in* rule assigning the treatment with the largest estimated treatment effect. We find that policy trees can capture the vast majority of treatment effect heterogeneity of the more flexible but less transparent and noninterpretable *plug-in* rule. Despite only moderate levels of heterogeneity, the policy tree yields a larger, albeit insignificant, increase in take-up than each individual treatment.

Our study relates to three fields of empirical research. First, sparked by methodological advances, especially the advent of causal forests (due to Wager & Athey, 2018), there is a burgeoning literature estimating heterogeneous treatment effects using machine learning (e.g., Davis & Heller, 2017; Knaus et al., 2022; Knittel & Stolper, 2021).[3] While studies in this literature emphasize the potential of estimating heterogeneous effects for improved targeting, they usually do not explicitly derive interpretable targeting rules. Second, we build on the expanding literature applying statistical decision rules. The vast majority of applied studies, including those discussed above (i.e., Assunção et al., 2022; Bhattacharya & Dupas, 2012; Frölich, 2008; Kitagawa & Wang, 2023; Lechner & Smith, 2007), only provide backtest results about the ex-post performance of policy targeting rules. Ida et al. (2022) propose a policy-learning framework that allows participants to self-select their treatment and apply their method to a residential energy rebate program. Closest to our study are Gerarden and Yang (2022) and Cagala et al. (2021). Gerarden and Yang (2022) follow the methodology of Kitagawa and Tetenov (2018) to estimate policy rules for a behavioral intervention targeted at reducing household electricity usage, but do not implement the derived policy rules. Similar to us, Cagala et al. (2021) consider policy trees in an application to maximizing fundraising and gauge the performance of the estimated policy tree on out-of-sample data. We add to this literature by fielding the estimated optimal policy rule in the second phase of our experiment, which allows us to directly evaluate the performance against other policy rules. Furthermore, both Cagala et al. (2021) and Gerarden and Yang (2022) focus on the choice between two treatment options, whereas we are concerned with the more challenging problem of multi-action policy learning. Third, we contribute to the larger literature on informational interventions aimed at increasing take-up of government services and subsidies among eligible people (e.g., Bhargava & Manoli, 2015; Finkelstein & Notowidigdo, 2019; Goldin et al., 2022; Hotard et al., 2019). Beyond contributing to these three strands, this article aims to make policy learning accessible to a wider audience by offering an introduction relevant both for randomized field experiments and applications relying on observational data.

This article proceeds as follows. Section 2 provides an introduction to policy learning. Section 3 turns to our application. We contextualize our application, describe the data, the treatments, and the study design in Sections 3.1–3.4. We summarize the results of the exploration and exploitation phase in Sections 3.5 and 3.6. Section 4 concludes the study.

2 | MULTI-ACTION POLICY LEARNING

In this section, we provide a brief review of (multi-action) policy learning, with a special focus on the policy learning framework of Zhou et al. (2022). While we rely on a randomized experimental design to learn the optimal policy rule in our application, we also discuss the setting where one has to rely on unconfoundedness assumptions, thereby illustrating the generality of the methodological framework.

The aim of policy learning is to formulate a policy rule $\pi(X)$ designed to maximize the expected value of Y, the outcome of interest. A policy rule assigns a treatment a from the choice set of treatment options $\mathcal{A} = \{1, 2, \dots, D\}$ to each individual based on their observed covariates X. Note that $\mathcal{A}$ may include the no-treatment option. Formally, $\pi(X)$ is a function mapping individual characteristics to one of the treatment options in $\mathcal{A}$. For example, a policy rule might assign treatment 1 to every person below age 30, treatment 2 to individuals aged 30–40, and treatment 3 to individuals older than 40.

[3]Other methods for estimating conditional average treatment effects using machine learning include Chernozhukov, Demirer, et al. (2018) and Künzel et al. (2019). For an overview, see Knaus et al. (2021) and Jacob (2021).

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2.1 | Estimating optimal policies

Before we turn to the estimation of optimal policies, it is instructive to consider a candidate policy rule $\pi'(X)$ and assess its effectiveness. We assume that we have access to the sample $\{Y_i, A_i, X_i\}$ for $i = 1, \dots, n$, which is drawn from the joint population distribution P. The sample data include the treatment received, A_i, the realized outcome, Y_i, as well as observed individual i's characteristics X_i. In our application, the data stem from the exploration phase of the randomized controlled trial, but the general approach also extends to observational data.

As typical in the causal effects literature, we assume the existence of the potential outcomes $\{Y_i(1), Y_i(2), \dots, Y_i(D)\}$, which are the outcomes if individual i had received treatments 1, 2, ..., D (Imbens & Rubin, 2015; Rubin, 1974). This allows us to define the expected reward of $\pi'(X)$, which is the expected value of the potential outcomes if the policy rule had been followed, that is, $Q(\pi'(X_i)) = E[Y_i(\pi'(X_i))]$ where $E[\cdot]$ denotes the expectation with respect to the population P. In nonexperimental settings, the fundamental challenge of estimating the reward of a candidate policy $\pi'(X)$ is that we only observe $Y_i = Y_i(A_i)$ and that individuals might self-select into treatment options that optimize their expected pay-off.

The offline policy learning literature commonly imposes the following set of assumptions[4]:

Assumption 1.

(a) Unconfoundedness: $Y_i(1), \dots, Y_i(D)\} \perp A_i | X_i$.
(b) Overlap: There exists some $\eta > 0$ such that $e_a(X_i) \geq \eta$ for any $a \in \mathcal{A}$ and X, where $e_a(X_i) \equiv P(A_i = a|X_i)$ is the (generalized) propensity scores for treatment a.
(c) Boundedness: The potential outcomes are contained on a finite interval in $\mathbb{R}^D$.

Under these assumptions, we can evaluate the reward of a candidate policy π' by taking the weighted average across observations that align with the candidate policy rule, that is,

$$\hat{Q}_{IPW}(\pi'(X_i)) = \frac{1}{n} \sum_{i=1}^{n} \frac{\mathbb{1}\left\{A_i = \pi'(X_i)\right\} Y_i}{e_{A_i}(X_i)}. \tag{1}$$

We inversely weight by the (generalized) propensity score $e_a(X_i)$ to allow for endogenous selection into treatment (Kitagawa & Tetenov, 2018; Swaminathan & Joachims, 2015).

Suppose that the policymaker considers a number of policy rules that depend on X_i, for example, $\Pi' = \{\pi'(X_i), \pi''(X_i), \pi'''(X_i)\}$ where Π' is the set of candidate policies.[5] The optimal policy is the policy in the candidate set Π' that maximizes the expected reward; formally, $\pi^\star(X_i) = \arg\max_{\pi \in \Pi'} Q(\pi(X_i))$. Accordingly, we can leverage our sample to estimate the optimal policy rule as

$$\hat{\pi}(X_i) = \arg\max_{\pi \in \Pi'} \hat{Q}_{IPW}(\pi(X_i)), \tag{2}$$

where $\hat{\pi}(X_i)$ is the policy from the set of candidate policies Π' that maximizes the estimated reward. The performance of policy learner $\hat{\pi}(X_i)$, which estimates $\pi^\star(X_i)$ from the data, is measured by its regret, $R(\hat{\pi}(X_i)) = Q(\pi^\star(X_i)) - Q(\hat{\pi}(X_i))$. The regret measures the difference between the reward of the unobserved optimal policy and the value of the estimated policy.

[4]The assumptions are standard in the causal inference literature (e.g., Imbens, 2004; Rosenbaum & Rubin, 1983) and have been adopted more recently in the literature on offline policy learning (e.g., Kitagawa & Tetenov, 2018; Zhou et al., 2022). Unconfoundedness in (a) states that we observe all necessary covariates, which allows us to account for selection biases. The overlap assumption in (b) uses the definition of the (generalized) propensity score $e_a(X_i)$, which generalizes the definition of propensity scores to accommodate multi-valued treatments (Imbens, 2000). Specifically, $e_a(X_i)$ denotes the propensity of taking up treatment a given observable characteristics X_i. The boundedness assumption in (c) serves the purpose of simplifying mathematical proofs but can be replaced by weaker assumptions (Zhou et al., 2022).

[5]Note that we leverage the same covariates X_i to adjust for selection effects in (1) and to form policy rules in (2). There may, however, be good reasons to use distinct covariate sets in each step. For example, legal or ethical concerns might mandate the exclusion of protected characteristics from the policy rule (e.g., gender, nationality, religion). Yet, the inclusion of these characteristics in the propensity score estimation could be necessary if prior evidence suggests their potential influence on treatment allocation. In randomized experiments, a consistent estimation of the reward does not require covariate adjustment but can enhance statistical precision.

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2.2 | Cross-fitting and double-robust estimation

If the propensity scores $e_a(X)$ are known, the regret converges to zero at $\sqrt{n}$-rate (Kitagawa & Tetenov, 2018; Swaminathan & Joachims, 2015). If the exact assignment mechanism is not known, which is typically the case in nonexperimental settings, we have to estimate $e_a(X)$ from the data. One approach is to estimate $e_a(X)$ using the full sample and plug the estimates into (1). However, this approach generally yields suboptimal convergence rates unless we impose strong convergence rate requirements on the first-step estimator (Kitagawa & Tetenov, 2018). The suboptimal performance can be attributed to the own-observation bias, which arises if the first-step estimation error from the propensity score estimation is correlated with the error associated with estimating the reward. To allow for a general class of data-adaptive nonparametric estimators, including popular supervised machine learners such as random forests, which are more robust towards unknown data structures, Zhou et al. (2022) combine two strategies for policy learning: cross-fitting and double-robust estimation using augmented inverse-propensity weighting (AIPW). We discuss each strategy in turn.

To illustrate how cross-fitting addresses the own-observation bias, consider the simple case where we randomly split the data into two subsamples, referred to as auxiliary and main samples. In the first step, we leverage the auxiliary sample for the estimation of conditional expectation functions (e.g., the propensity scores). The second step uses out-of-sample predicted values from the first step on the main sample to estimate the reward. This sample-splitting approach resolves the own-observation bias since the second step is, after conditioning on the auxiliary sample, independent from the first-step estimation error.

Cross-fitting extends this sample-splitting approach by flipping the auxiliary and main samples, thus effectively using the full sample for both the first- and second-step estimation. Cross-fitting also allows the sample to be split into more than two partitions.[6] Specifically, to implement cross-fitting, we randomly split the sample into K folds of approximately equal size. We use $\hat{e}_a^{-k(i)}(X_i)$ to denote the *cross-fitted* (generalized) propensity score of observation i for treatment a. The cross-fitted predicted value is calculated as the out-of-sample predicted value from fitting an estimator on all folds but fold $k(i)$, which is the fold that observation i falls into. Similarly, we introduce $\hat{\mu}_a^{-k(i)}(X_i)$ which is the cross-fitted predicted value of the outcome under treatment a using predictors X_i, that is, it is a cross-fitted estimate of $\mu_a(X_i) \equiv E[Y_i(a)|X_i]$.

Double robust estimation of the reward allows for nonrandom treatment assignment under uncounfoundedness. The estimator adjusts for biases arising from selective treatment allocation by combining the reweighting approach of inverse-propensity weighting as used in (1) with outcome adjustment (as used in regression-based adjustment). The advantage over the IPW estimator is that the resulting double-robustness property guarantees consistency if either the propensity scores $e_a(X_i)$ or the conditional expectation of outcome given covariates, that is, $\mu_a(X_i)$, are correctly specified. Using the cross-fitted estimates $\hat{e}_a^{-k(i)}(X_i)$ and $\hat{\mu}_a^{-k(i)}(X_i)$, we can define the cross-fitted AIPW (CAIPW) estimator of the reward as[7]

$$\hat{Q}_{CAIPW}(\pi(X_i)) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{\mu}_{A_i}^{-k(i)}(X_i)}{\hat{e}_{A_i}^{-k(i)}(X_i)}\mathbb{1}\{A_i = \pi(X_i)\} + \hat{\mu}_{\pi(X_i)}^{-k(i)}(X_i)\right). \tag{3}$$

The first term in (3) adjusts the observed outcome by subtracting the conditional expectation of the outcome under the observed treatment and by inversely weighting with the propensity scores if the observed treatment, A_i, is equal to the treatment recommended by the policy, $\pi(X_i)$. The second term adds the conditional expectation of the outcome under the treatment assigned by the policy. Using the double-robust estimator of the reward, we can estimate the optimal policy by evaluating $\hat{Q}_{CAIPW}(\pi(X_i))$ for all candidate policies in Π', that is, we calculate $\hat{\pi}(X_i)_{CAIPW} = \arg\max_{\pi\in\Pi'}\hat{Q}_{CAIPW}(\pi(X_i))$.

2.3 | Policy class

So far, we have assumed a predefined set of candidate policies. In many applications, however, we wish to learn policies flexibly from the data instead of relying on a predefined set of policy rules. A fully flexible approach could assign each individual to the treatment for which the estimated treatment effect is the largest. This *plug-in policy rule* requires no functional form restrictions but may be inappropriate when stakeholders wish to learn about the drivers of treatment efficacy and have hesitations to rely on a black-box treatment assignment mechanism.[8]

[6]The causal machine learning literature frequently relies on sample splitting approaches, such as cross-fitting; see for example Chernozhukov, Chetverikov, et al. (2018) for the estimation of average treatment effects and Wager and Athey (2018) for the estimation of CATE using causal forests.

[7]The function $\mathbb{1}\{\cdot\}$ denotes the indicator function.

[8]For formal results on plug-in rules, see Hirano and Porter (2009) and Bhattacharya and Dupas (2012).

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

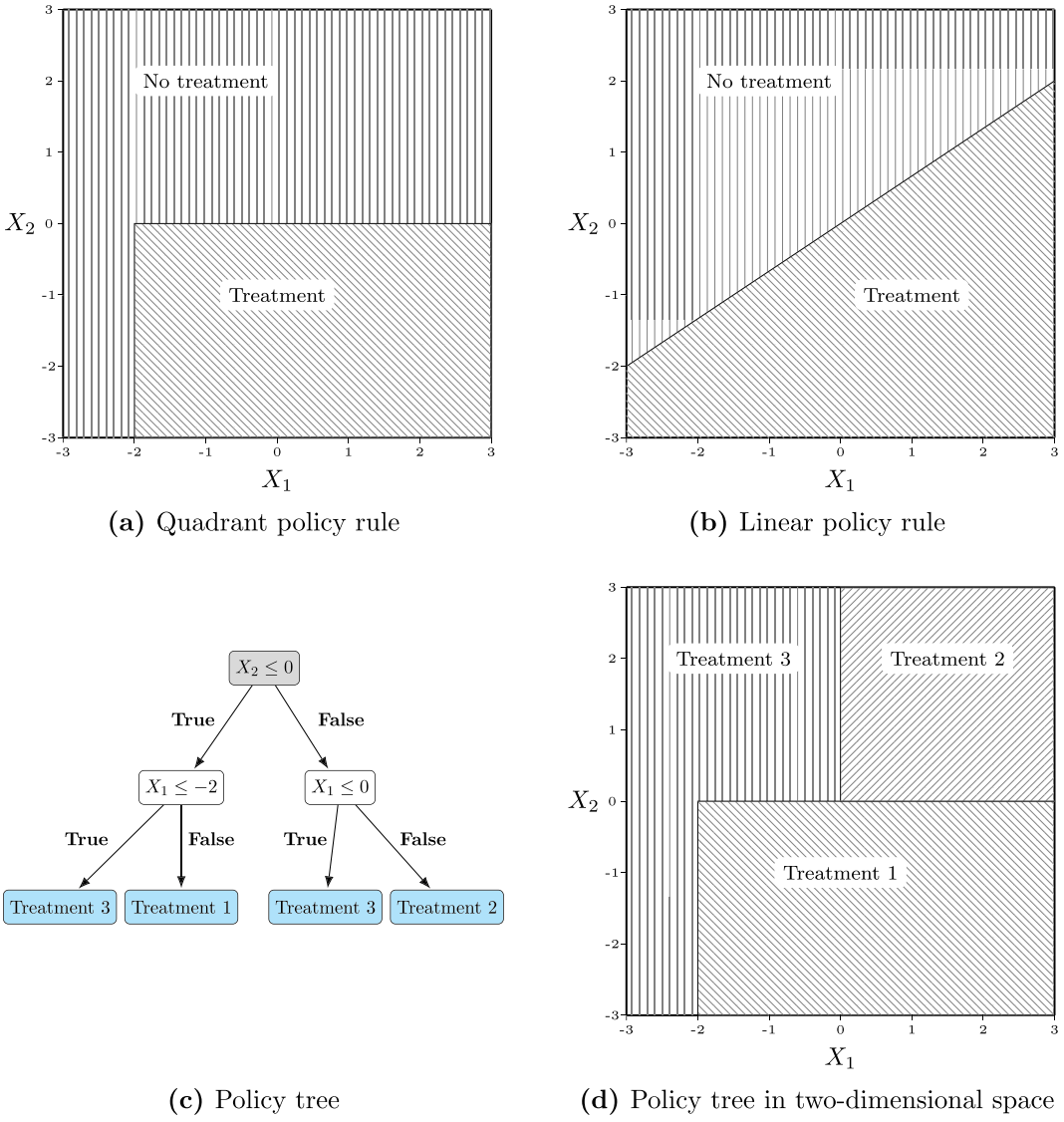


FIGURE 1 Illustrative examples of policy rules with different functional forms. *Note:* The figure shows three examples of policy rules that assign treatments based on two covariates X_1 and X_2. Panels (a) and (b) illustrate the quadrant and linear policy rule considered in Kitagawa and Tetenov (2018) for the case of a binary treatment. The quadrant rule is given by $\pi(X_i) = \mathbb{1}\{s_1(X_{1i} - \beta_1) \geq 0\}\mathbb{1}\{s_2(X_{2i} - \beta_2) \geq 0\}$. The linear policy rule is defined by $\pi(X_i) = \mathbb{1}\{\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} \geq 0\}$. Panels (c) and (d) provide two alternative illustrations of the same policy tree with three treatment options.

Policy learning allows for the estimation of interpretable treatment rules from the data. To this end, we must choose a suitable policy class from which we estimate the optimal policy. Figure 1 illustrates several options. Figure 1a,b shows examples of the quadrant and linear policy rules considered by Kitagawa and Tetenov (2018) in an application to active labor market programs with a binary treatment and two covariates. Another possible policy class is given by trees. Trees are widely employed as predictive tools that construct predictions by splitting the feature space optimally into nonoverlapping regions. In the prediction context, classification and regression trees yield the same prediction for observations falling into the same region. In the policy context, observations falling into the same region are assigned the same treatment action. Athey and Wager (2021) suggest using trees for policy learning, and Zhou et al. (2022) generalize policy trees to settings where the treatment is multivalued.[9] Figure 1c shows an example of a multi-action policy tree, and Figure 1d visualizes the same tree in a two-dimensional space.

[9]Zhou et al. (2022) describe how the optimization of policy trees can be regarded as a mixed integer program.

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

3 | PERSONALIZING NATURALIZATION CAMPAIGNS

In this section, we apply the multi-action policy tree to an information campaign encouraging eligible immigrants residing in the City of Zurich to apply for Swiss citizenship. We introduce the policy context in Section 3.1 and discuss data and treatments in Sections 3.2 and 3.3. Section 3.4 summarizes the study design and estimation methodology. Results are presented in Sections 3.5 and 3.6.

3.1 | Background: Immigrant integration and citizenship

The integration of immigrants into the host-country fabric and economy is a central policy issue in many countries across the globe. One promising policy to foster integration is naturalization, that is, the process of awarding host-country citizenship to immigrants (Dancygier, 2010; Goodman, 2014). Observational studies relying on difference-in-difference models and regression discontinuity designs comparing similar naturalized and nonnaturalized immigrants show that acquiring host-country citizenship can positively impact the integration of immigrants by increasing their earnings and labor market attachment (Gathmann & Keller, 2018; Gathmann & Garbers, 2023; Govind, 2021; Hainmueller et al., 2019; Mazzolari, 2009; OECD, 2011; Vink et al., 2021), fostering political efficacy and knowledge (Hainmueller et al., 2015), spurring cultural assimilation and cooperation (Felfe et al., 2021; Keller et al., 2015), and reducing feelings of isolation and discrimination (Hainmueller et al., 2017).[10] This process can also benefit the host society by increasing immigrants' contributions to economic growth, lowering their dependency on welfare, and, by extension, reducing societal tensions and strengthening social cohesion (for reviews, see National Academies of Sciences, et al., 2016; Pastor & Scoggins, 2012).

Despite these potential benefits, naturalization rates remain low in many countries (Blizzard & Batalova, 2019). What explains this mismatch between the benefits of host-country citizenship and the low demand for naturalization? Previous evidence from surveys and qualitative studies suggest that uncertainty about the eligibility criteria such as residency and language requirements can prevent immigrants from applying (Bauböck et al., 2006; Gonzalez-Barrera et al., 2013). Other studies highlight that—particularly in hostile immigration environments—a lack of encouragement by politicians, public administration, or the general public might deter immigrants (Bauböck et al., 2006; Bloemraad, 2002; Bloemraad et al., 2008). Furthermore, in earlier research using a tailored survey, we find evidence for informational deficits and the feeling that an application is not welcome by the host society (Hangartner et al., 2023). Lastly, in countries that unlike Switzerland do not allow for dual citizenship, immigrants might not be willing to give up the passport from their origin country to obtain host-country citizenship.[11]

To boost naturalization rates, countries, states, and municipalities across Europe and the USA have begun to turn to information campaigns to overcome hurdles to citizenship acquisition for eligible immigrants. While the content and scope of these naturalization campaigns vary, they often combine information provision about the naturalization process and requirements with an encouragement to apply for citizenship. Yet, despite the growing popularity of these campaigns across Europe and the USA, there exists little experimental research to evaluate its effectiveness. An important exception is Hotard et al. (2019), who show that a low-cost nudge informing low-income immigrants about their eligibility for a fee waiver increased the rate of citizenship applications by 8.6 percentage points (from 24.5% in the control group to 33.1%). Most similar to our study is Hangartner et al. (2023), who evaluated previous versions of the naturalization campaign of the City of Zurich and showed that a similarly low-cost letter (about CHF 1.20 per person; see below) combining information and encouragement increased naturalization rates by about 2.5 percentage points (from 6.0% in the control group to 8.5%).

Past naturalization campaigns, including the one by the City of Zurich mentioned above, have typically relied on a one-size-fits-all approach—despite the substantial diversity of the immigrant population in terms of, for example, country of origin, language skills, and age. There are good reasons to suspect treatment effect heterogeneity along various dimensions: Immigrants' willingness to naturalize and their susceptibility to certain information letters might depend on their

[10]With the exception of Mazzolari (2009), who studies immigrants from Latin American countries in the USA, the studies referenced above focus on France, Germany, and Switzerland. This might limit external validity since we expect the benefits of naturalization to be context-dependent and generally decline with lower naturalization hurdles. While testing this hypothesis requires more comparative research, Vernby and Dancygier (2019) provide initial evidence from a correspondence test varying citizenship in fictitious applications in Sweden that is consistent with this conjecture.

[11]Whether a person is allowed to retain the previous citizenship when naturalizing in another country generally depends on the regulations of both the origin and host country. Switzerland has guaranteed the right to hold dual (or more) citizenship without restrictions since 1992. Hence, eligible immigrants seeking Swiss citizenship are subject only to restrictions of their origin countries. Among the 10 largest countries by nationality in our sample (which jointly amount to 72% of origin countries), only Austria and Spain generally do not allow for dual citizenship.

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

current nationality due to the specific dual citizenship regulations, the relative benefits in terms of visa requirements *vis-á-vis* third countries, the attachment to the home country, and the attitudes of native citizens towards specific immigrants groups. For example, immigrants who feel discriminated against might be more likely to be persuaded by a letter welcoming them to set roots and apply for citizenship in their host country. Furthermore, language requirements might be less of a concern for immigrants who speak the same language (such as Austrians and Germans in Switzerland). Thus, tailoring such campaigns to the specific needs of diverse immigrants promises to deliver both a deeper understanding of the different hurdles that immigrants face and to increase the effectiveness of the campaign.

3.2 | Data

We draw our data from administrative sources of the Canton of Zurich. The data include records of whether and when eligible immigrants submit an application for Swiss citizenship to the City of Zurich during the study period, which allows us to define the outcome variable of our analysis. The data also include additional covariates which we use to identify and leverage treatment effect heterogeneity. These covariates are age, gender, nationality, years of residency in Switzerland, and years of residency in Zurich. The data also include an address identifier which allows us to assign the treatment on a building level to minimize contamination by spill-over effects.

The study sample includes all immigrants in the City of Zurich who satisfy the following criteria:

1. They were born on or before June 30, 2003 (i.e., they must have been at least 18 years of age at the start of the study),
2. They arrived in Switzerland on or before June 30, 2011,
3. They arrived in Zurich City on or before June 30, 2019,
4. They must have possessed a permanent residence permit (C permit) at the time of randomization (August 2021), and
5. They must not have received any information or encouragement letter in the past.

The first criterion ensures that only adults are in the study. Criteria 2–4 ensure that the entire sample meets the current residency and permit requirements for citizenship. The sample includes 5145 individuals.

3.3 | Treatment letters

Combining insights from the existing literature and our own surveys, we identify three key barriers to naturalization: (i) perceived complexity of the naturalization process, (ii) perceived difficulty of and uncertainty about naturalization requirements, and (iii) perception that naturalization is not welcome. In collaboration with the City of Zurich, we developed three treatment letters, each of which puts emphasis on one of the hurdles. Each treatment involves the receipt of a letter sent by representatives of the City of Zurich. The treatments differ in the sender, content, wording and design of the letters. The per-unit costs of the three treatments range between 1.20 and 1.50 CHF and are thus negligible compared with the fiscal benefits of naturalization.[12] We chose to develop distinct letters to keep the letters brief and understandable, thus avoiding the risk of an informational overload (Haaland et al., 2023). The letters, including enclosed flyers, were written in German. Appendix A.2 in the supporting information contains copies of the original letters in German as well as an English translation.

The *Complexity letter* consists of a short informational cover letter written by the City Clerk of the City of Zurich (see Appendix A.2.1) and a flyer. The half-page cover letter informs recipients that they meet the basic requirements for Swiss citizenship and directs them to sources of further information about the citizenship application process. The flyer included in the *Complexity letter* (shown in Figure A.2.2 in the supporting information) attempts to tackle the perceived complexity of the naturalization process. The left-hand side of the flyer shows a video screenshot and a QR code that directs readers to the video, explaining the naturalization process in a simplified way. The right-hand side encourages readers to scan another QR code redirecting to the contact and advice webpage[13] of the City of Zurich's citizenship office.

[12]Hainmueller et al. (2019) quantify the long-term effect of naturalization on immigrants' earnings at CHF 4,500 per year, which implies an increase in tax revenues for Swiss municipalities of at least CHF 450 per year.

[13]The first QR code redirects to https://www.stadt-zuerich.ch/portal/de/index/politik_u_recht/einbuergerungen.html (last accessed on December 7, 2022). The second QR code redirects to https://www.stadt-zuerich.ch/portal/de/index/politik_u_recht/einbuergerungen/kontakt-und-beratung.html (last accessed on December 7, 2022).

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The *Requirements letter* includes the same short informational cover letter as the *Complexity letter* but uses a different flyer addressing the perceived difficulty of the naturalization process (see Appendix A.2.3 in the supporting information). This flyer is also divided into two sections, each containing a descriptive text and a QR code. The QR code on the left-hand side redirects to the targeted, free-of-charge mobile application, which allows immigrants to study for the civics exam and test their knowledge with practice questions.[14] The section on the right lists the German language requirements for citizenship and the QR code redirects to a webpage containing more detailed information on the language requirements, exam costs, as well as a link to a practice language exam.[15]

The *Welcome letter* is an information and encouragement letter signed by the Mayor of the City of Zurich. The *Welcome letter* attempts to tackle the hurdle stemming from the perception that naturalization is not welcome (Hainmueller & Hangartner, 2013). The letter includes only a cover letter (shown in Appendix A.2.4 in the supporting information) that is a little less than one page long and contains three sections. The first section informs recipients that they meet the basic eligibility requirements for Swiss citizenship. The second section encourages them to play an active part in Zurich's political life by becoming a citizen. The last section briefly directs to sources for further information about the citizenship application process and states that the City hopes to see them at the next ceremony for new citizens. Hence, compared with the other two treatment letters, this letter puts more emphasis on the emotional and psychological aspects associated with naturalization and only provides minimal information.

3.4 | Experimental design and estimation methodology

This section summarizes the preregistered experimental design, estimation methodology, and evaluation strategy.[16] In the exploration phase of the project, we randomly divide the sample of 5145 eligible immigrants into two groups: Group A (60% of the sample) receives one of three treatment letters at random from the City of Zurich in October 2021, while Group B (40%) received no letter. The randomization design allocates one of the three treatment letters to individuals in Group A by building address and applied block randomization by nationality groups. The randomization by building address reduces the risk of spill-over effects among eligible immigrants living in the same or neighboring households. The block randomization by nationality group ensures that we have a roughly equal share of nationalities in Group A (including each subgroup receiving different letters) and Group B. We block on nationality groups given the importance of this effect moderator in earlier studies (Ward et al., 2019). The letters for this first wave were delivered on October 8, 2021.

The first-wage application outcomes enable us to estimate the average treatment effect of treatment letter d, that is, $E[Y_i(d) - Y_i(0)]$, and the conditional average treatment effect $E[Y_i(d) - Y_i(0)|X_i]$ where we use Y_i to denote the application outcome recorded at the end of March 2022 and $Y_i(d)$ its potential outcome under treatment d. The covariates X_i are country group of nationality, age, gender, years lived in Zurich, and years lived in Switzerland, which are constant over the sample period. We employ causal forests due to Wager and Athey (2018) and Athey et al. (2019), a nonparametric method for the estimation of heterogeneous treatment effects relying on random forests.

The main objective, however, is to leverage the first-wave application outcomes Y_i and individual characteristics X_i to fit a multi-action policy tree based on the estimation methodology of Zhou et al. (2022) outlined in Section 2. We opt for policy trees as they easily generalize to the multi-action settings. Furthermore, policy trees are transparent and simple to interpret, even without statistical training, making them attractive in a public policy context where users of the research vary in statistical literacy and often view black-box methods with skepticism. To select the tree depth, we consider a validation exercise: In each iteration, we randomly split the wave-1-data (including untreated) into training and test data with a 60/40 split and sample from each partition separately with replacement to construct bootstrapped training and validation data sets of sizes $n_1 = 4871$ and $n_2 = 1857$. We then fit a policy tree on the bootstrapped training data and estimate the difference in reward between alternative policy rules on the bootstrapped validation data.

[14]The mobile application is developed by the City of Zurich and named *Einbügerungstest Code Schweiz*, which translates to Naturalization Test Code Switzerland.

[15]The website, which the QR code redirected to, moved to https://www.stadt-zuerich.ch/portal/de/index/politik_u_recht/einbuergerungen/kenntnisse/sprachlicheanforderungen.html on October 21, 2022, due to a mistake by the website maintainers. As a consequence, the QR code broke more than 5 months after the letter was dispatched to wave two participants. We show in Table A.4 in the supporting information, where we only consider the naturalization applications recorded up to 5 months after letter dispatch, that our main results in Table 3 are not affected by this issue. We thus use, in line with the pre-analysis plan, application outcomes recorded 7 months after letter dispatch in the remainder of the study.

[16]The study was preregistered online (https://osf.io/9wf4t).

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

In the exploitation phase, we field the fitted policy tree on not-yet-treated individuals in Group B. Specifically, in order to evaluate the performance of the policy rule, we randomly subdivide Group B into two subgroups, referred to as Group B.1 and Group B.2, and send treatment letters to Group B.1 based on the estimated policy rule, while Group B.2 receive a random treatment letter (with one-third probability for each letter). We randomize by building address for the random division into Groups B.1 and B.2, as well as for the randomization of treatments within Group B.2. The City of Zurich delivered the letters for the exploitation phase on May 6, 2022.[17]

The evaluation compares the policy tree against no treatment, random treatment allocation, and conventional one-size-fits-all policy rules that always assign the same treatment to everyone, ignoring treatment effect heterogeneity. To this end, we estimate models of the form:

$$Y_{it} = W_{it}'\beta + f(X_i, \delta_t) + \varepsilon_{it} \tag{4}$$

where Y_{it} is the application outcome of eligible immigrant i at the end of wave $t \in \{1, 2\}$. We add the wave subscript t to accommodate the two-wave structure of the data. The outcomes for the evaluation analysis were recorded approximately 7 months after the date of letter dispatch t.[18] The time-invariant covariates X_i are defined above. δ_t is a dummy for wave $t \in \{1, 2\}$ and accounts for seasonal effects and other external shocks that may affect application rates. The vector W_{it} assigns individuals to treatment groups and is defined as $W_{it} = (Letter^1_{it}, Letter^2_{it}, Letter^3_{it}, Nothing_{it}, PolicyTree_{it})$ or $W_{it} = \left(Random_{it}, Nothing_{it}, PolicyTree_{it}\right)$, respectively, where $Letter^j_{it}$ is set to 1 if the individual i was randomly assigned to treatment letter $j \in \{1, 2, 3\}$ for wave t, 0 otherwise. $Nothing_{it}$ is set to 1 if the individual i has received no treatment in wave t, and $PolicyTree_{it}$ equals 1 if individual i has received the treatment letter assigned to them by the policy tree. Finally, $Random_{it}$ is set to 1 if individual i was randomly assigned to one of the three letters, 0 otherwise.

We estimate (4) by linear regression using only the elementary controls but also consider more flexible methods. Namely, we use post-double selection lasso (PDS Lasso Belloni et al., 2014) and double-debiased machine learning (DDML; Chernozhukov, Chetverikov, et al., 2018) where we extend the set of controls by interaction terms and second-order polynomials.[19] We cluster standard errors by building addresses, that is, the level at which the treatment was applied.[20]

3.5 | Results from the exploration phase: Learning the policy rule

We begin by analyzing the results from the exploration phase of the experiment using naturalization applications received by the end of March 2022 (i.e., wave 1). Descriptive statistics of the wave-1-data are provided in Table 1. We proceed in three steps: estimation of (conditional) averages of treatment effects, tuning policy trees using a validation exercise, and fitting the policy tree on the full wave-1-data.

First, we fit a multi-arm causal forest to estimate average treatment effects, as well as conditional average treatment effects by nationality group and years lived in Switzerland (Athey et al., 2019; Wager & Athey, 2018). Results are displayed in Figure 2.[21] The average treatment effects for the first-wave sample imply that the *Complexity letter* increases application rates by 1.08 p.p. (*s.e.* = 0.91), the *Requirements letter* by 4.33 p.p. (*s.e.* = 1.04), and the *Welcome letter* by 3.51 p.p. (*s.e.* = 1.03), relative to the control condition of no letter.[22]

[17]Note that for practical reasons, there was a two-month time gap between measuring the application outcomes in March 2022 and sending out the letter in May.

[18]The application outcomes for the evaluation analysis were recorded in May 9 and December 9, 2022, respectively. In Table A.4 in the supporting information, we provide alternative results where we consider all application outcomes until March 21 and October 21, 2022, respectively (see fn. 15).

[19]For the Post-Double Selection Lasso, we use cluster-robust penalty loadings of Belloni et al. (2016). With regard to DDML, we use 10 cross-fitting folds, 5 cross-fitting repetitions and use stacking with a set of candidate learners including linear regression, lasso, ridge, random forests and gradient boosting (Ahrens et al., 2024)

[20]We note that the clustered standard errors do not account for sampling variability arising from the estimation of the policy rules. The issue is akin to the well-known generated regressor problem which occurs when a regressor is unobserved and replaced by a first-step estimate. The generated regressor problem is usually addressed using standard-error adjustments Pagan (1984) and Murphy and Topel (1985) or, most commonly, using bootstrapping; see e.g. review in Chen et al. (2023) or Wooldridge (2010). Neither of these approaches is feasible in our setting. Analytical standard errors are, to our knowledge, not available for this specific problem. Bootstrapping or other resampling techniques would require us to repeatedly field policy rules fitted on bootstrapped samples of the data in order to capture the variability in estimated policy rules, which is practically infeasible. We thus interpret the standard errors with caution.

[21]We removed 274 individuals who moved between October 2021 and March 2022, resulting in an estimation sample of 4871 individuals.

[22]See Hangartner et al. (2023) for a discussion the letters' efficacy in overcoming specific hurdles.

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 Descriptive statistics of wave-1 data.

	Avg.	*St.dev.*	*Min*	*Max*	*Obs.*
Dependent variable:					
Naturalization application	0.07	0.25	0.00	1.00	4871
Covariates:					
Age	41.79	10.95	19.00	99.00	4871
Female	0.46	0.50	0.00	1.00	4871
Years in Switzerland	14.88	7.44	11.00	67.00	4871
Years in Zurich	9.08	4.06	3.00	20.00	4871
Regions:					
Americas & Caribbean	0.05	0.22	0.00	1.00	4871
Asia	0.06	0.25	0.00	1.00	4871
Central-East Europe	0.04	0.19	0.00	1.00	4871
Germany and Austria	0.37	0.48	0.00	1.00	4871
Italy	0.10	0.30	0.00	1.00	4871
Middle Easy and Northern Africa	0.01	0.12	0.00	1.00	4871
South-East Europe	0.12	0.32	0.00	1.00	4871
Spain and Portugal	0.12	0.33	0.00	1.00	4871
Stateless	0.00	0.04	0.00	1.00	4871
Sub-Saharian Africa	0.02	0.12	0.00	1.00	4871
Western Europe	0.11	0.32	0.00	1.00	4871

Note: The table shows summary statistics for covariates and dependent variables measured until March 2022.

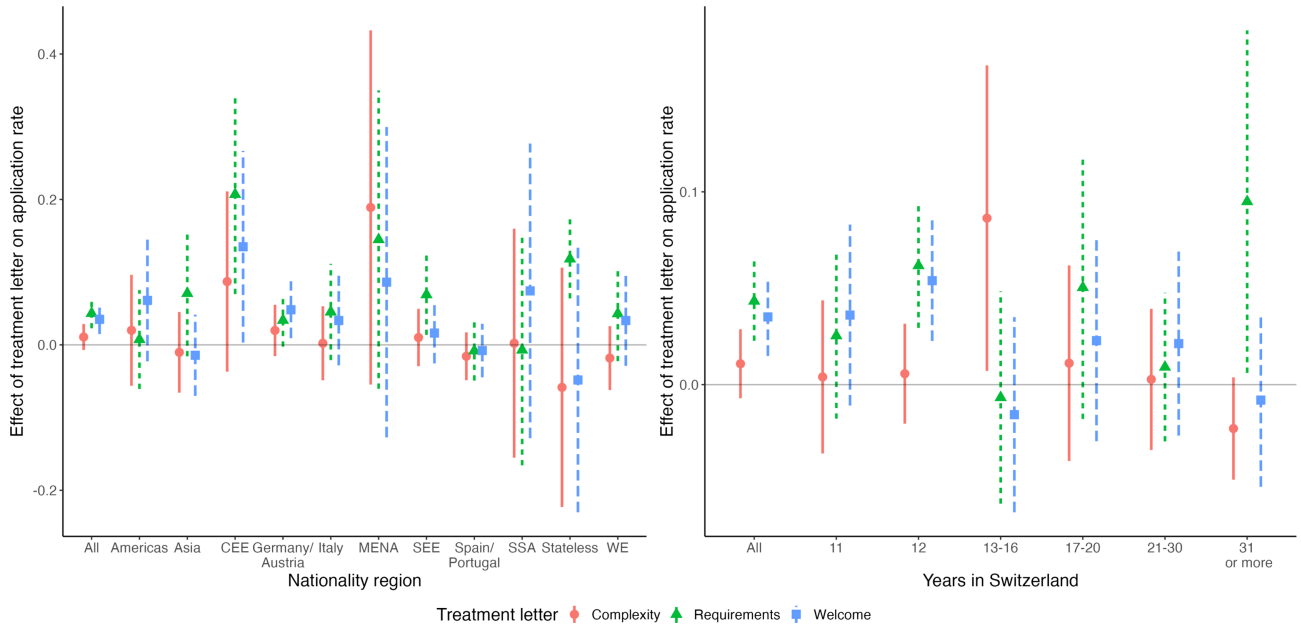


FIGURE 2 Average and conditional average treatment effects. *Note:* The figures shows the average and conditional average treatment effects by group where the groups are formed based on nationality and years of residence in Switzerland. The regions are the Americas, Asia, Central and East Europe (CEE), Germany and Austria, Italy, Middle East and Northern Africa (MENA), South-East Europe, Spain and Portugal, Stateless and Sub-Saharian Africa (SSA). The treatment effects are estimated using a multi-arm causal forest and using the R package `grf` (Athey et al., 2019; Tibshirani et al., 2022; Wager & Athey, 2018).

The left panel of Figure 2 shows only moderate heterogeneity in treatment effects by nationality. The *Welcome letter* appears to have slightly stronger effects for immigrants from Germany and Austria, consistent with the idea that Germans and Austrians do not perceive complexity and difficulty as major hurdles due to their cultural proximity and language. At the same time, the *Welcome letter* is also the most effective letter for immigrants from the Americas, which could indicate that this minority group does not feel very welcome in Switzerland. The relative effect size of the *Requirements letter* is particularly large for immigrants from Central-Eastern and South-Eastern Europe, as well as for stateless immigrants. The right panel of Figure 2 indicates that the *Complexity letter* has the largest effect on application rates among eligible

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 Estimated reward of the policy rule compared to randomization, always the same treatment and no treatment.

	One-size-fits-all			Random	Policy tree			Plug-in
	Complexity	**Requirem.**	**Welcome**	**treatment**	$d = 2$	$d = 3$	$d = 4$	**rule**
Nothing	0.917	4.096★ ★ ★	3.245★ ★ ★	2.755★ ★ ★	5.369★ ★ ★	5.545★ ★ ★	5.436★ ★ ★	6.396★ ★ ★
	(1.007)	(1.092)	(1.106)	(0.761)	(1.015)	(0.985)	(0.936)	(0.943)
Always 1		3.180★★	2.329★	1.838★★	4.452★ ★ ★	4.628★ ★ ★	4.520★ ★ ★	5.479★ ★ ★
		(1.295)	(1.291)	(0.736)	(1.174)	(1.115)	(1.080)	(1.071)
Always 2			−0.851	−1.342	1.273	1.448★	1.340★	2.299★ ★ ★
			(1.351)	(0.772)	(0.795)	(0.791)	(0.780)	(0.748)
Always 3				−0.491	2.124★★	2.300★ ★ ★	2.191★ ★ ★	3.150★ ★ ★
				(0.777)	(0.860)	(0.865)	(0.827)	(0.846)
Random					2.614★ ★ ★	2.790★ ★ ★	2.682★ ★ ★	3.641★ ★ ★
					(0.590)	(0.552)	(0.502)	(0.489)
Policy tree ($d = 2$)						0.176	0.067	1.027★ ★ ★
						(0.296)	(0.304)	(0.286)
Policy tree ($d = 3$)							−0.109	0.851★ ★ ★
							(0.246)	(0.248)
Hybrid tree ($d = 2$)								0.959★ ★ ★
								(0.221)

Note: The table reports the difference in estimated rewards between policy rules based on wave-1 data (including untreated immigrants of Group B). Specifically, each cell corresponds to the gain in reward of a specific policy rule (shown in columns) relative to alternative policy rules (listed in rows). The results are based on a resampling exercise where we randomly split the wave-1 data into training and test data using a 60/40 split and separately draw $n_1 = 4871$ and $n_2 = 1857$ observations with replacement from the training and test data. We use 500 repetitions and report the average difference in rewards and associated bootstrapped standard errors.***0.01. Significance level.**0.05. Significance level.*0.1. Significance level.

immigrants who have lived between 13 and 16 years in Switzerland. In contrast, eligible immigrants who have lived for more than 30 years in Switzerland are especially receptive to the requirements letter, suggesting that the perceived difficulty of the naturalization process may discourage some eligible immigrants from applying over long periods. This effect may also be partially driven by age since we also find the *Requirements letter* to have the largest effect among immigrants aged 46 and above (see Figure A.4 in the supporting information appendix). Finally, we find that men are slightly more receptive to the letter treatments overall than women, but the ranking of treatment letter efficacy is the same (see Figure A.4 in the supporting information).

Second, we conduct the validation exercise outlined above to assess the out-of-sample performance of various policy rules and to select the tree depth of the policy tree. We focus on policy trees with tree depths of 2 and 3. We also estimate a hybrid policy tree of depth 4. Hybrid policy trees rely on a computationally less costly but approximate optimization algorithm (Sverdrup et al., 2022). For comparison, we consider (i) one-size-fits-all rules that always assign one of the *Complexity*, *Requirements* or *Welcome* letters, (ii) random allocation of one of the three letters, and (iii) a model-free plug-in rule that assigns the treatment for which the estimated reward is the largest. We repeat the exercise 500 times and report average differences in rewards and bootstrapped standard errors in Table 2.[23] The table reports in each column the gain in reward of a specific policy choice compared to alternative policy rules (shown in rows). For instance, the coefficient of 0.917 ($s.e. = 1.007$) in the top-left entry corresponds to the gain in reward of a one-size-fits-all policy rule assigning the *Complexity letter* to everyone relative to a policy rule assigning no letter. We find that all three policy trees outperform each individual treatment letter as well as random treatment allocation. Among the three policy trees, the tree of depth 3 performs marginally better than trees of depths 2 and 4. As expected, the plug-in rule shows overall the best performance. However, the plug-in rule provides no insights into the drivers of treatment effects. The results thus highlight the trade-off between interpretability and performance but also show that, in this context, the best-performing policy tree is able to reach more than 85% of the performance of the plug-in rule.

Third, in light of the advantages and limited costs of policy trees in this setting, we opted for implementing the policy tree of depth 3. Following the approach of Zhou et al. (2022) as outlined in Section 2, we trained the policy tree on wave 1 data, including Group A (who received a letter in the first wave) and Group B (who did not receive a letter in the first wave). Since we randomized treatment assignment in the first wave, we did not need to estimate the propensity scores but

[23]We opted for this approach rather than K-fold cross-validation as it allows us to match the sizes of the training and validation data to the actual sample sizes. However, we obtain similar results when applying K-fold cross-validation.

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

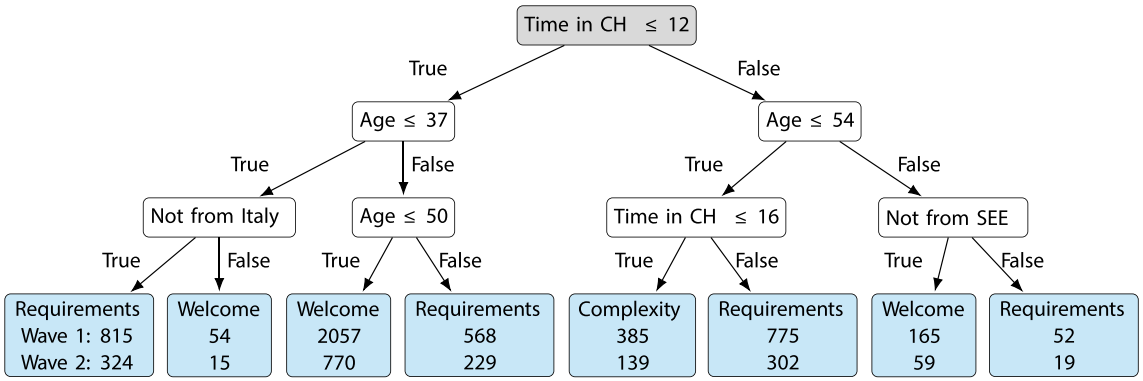


FIGURE 3 Fitted policy tree. *Note:* The figure shows the policy tree fitted to data from wave 1. The size of the training sample is 4871. The numbers at the bottom indicate the number of individuals assigned to each terminal node in the training sample and in Group B.

plugged the known treatment shares into (3).[24] We used multi-arm causal forests to estimate the double robust scores, although other estimators are possible. The fitted policy tree $\hat{\pi}$ of depth three is displayed in Figure 3. The boxes at the bottom of the tree show the assigned treatment for the wave-1 sample and the wave-2 sample (i.e., Group B) per terminal node. For instance, the very-left branch assigns individuals who have spent no more than 12 years in Switzerland are aged 37 years or younger and who are not from Italy to the requirements treatment. A total of 815 individuals in total and 324 individuals from Group B fall into that category. In total, 139 individuals of Group B are assigned to the *Complexity letter*, 874 individuals to the *Requirements letter*, and 844 to the *Welcome letter*.[25] The splits in the tree are based on years in Switzerland, age, and only two nationality indicators, but no split is based on gender confirming that the relative performance of each letter is the same for women and men. It is also noteworthy that no individuals were assigned to receive no letter, which suggests that at least one of the three letters has a positive effect for every individual.

3.6 | Results from the exploitation phase: Evaluating the policy rule

Table 3 shows the results of the evaluation based on estimating versions of (4) using ordinary least squares (OLS; see columns 1–3), PDS lasso (columns 4 and 5), and DDML (columns 6 and 7). The sample includes only wave 2 in column 1 and both waves in the remaining columns. The reference group in column 1 is random treatment allocation, while the base group in columns 2–7 is no treatment. Panel A reports the coefficient estimates, and Panel B compares the policy rule using policy trees against each individual treatment letter and random treatment allocation.

According to the OLS results in columns 1–3, the treatment assignment by policy tree increased the application rate by 1.79 (*s.e.* = 1.36) to 1.90 p.p. (1.36) relative to random treatment and by around 5.13 p.p. (1.61) compared with no treatment. Random allocation is associated with an application rate increase of approximately 3.23 p.p. (0.82). Turning to the individual treatments, we find that the *Welcome letter* yields overall the largest increase in application take-up with an effect size around 3.79 p.p. (1.07), closely followed by the *Requirements letter* with an effect size around 3.65 p.p. (1.10). The *Complexity letter* performs substantially worse in comparison, with an effect size of 2.23 (*s.e.* = 1.04). Panel B shows that the policy tree performs better than random treatment or each individual treatment option. The take-up increase compared with the best-performing individual treatment (the *Welcome letter*) is 1.03 p.p. but statistically insignificant. The PDS lasso estimates are almost identical, and the DDML estimator yields effect sizes only marginally smaller.[26]

[24]We note that in a setting where $e_a(X_i)$ is known, the IPW estimator of the reward in (1) is also applicable. We find in simulations that the AIPW estimator using the known propensity scores outperforms the IPW estimator.

[25]We assigned policies for Groups B.1 and B.2 after removing individuals who either applied without being treated (99 individuals) or moved out of the municipality of Zurich (101 individuals).

[26]Appendix Table A.5 in the supporting information also shows alternative results using logistic regression. The average marginal effects from logistic regression are almost identical to those from OLS.

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 3 The effect of the policy rule compared to randomization, always the same treatment and no treatment.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
	Dependent variable: Naturalization application						
Panel A. Coefficient estimates							
Policy tree	1.794	5.124***	5.127***	5.004***	5.005***	4.702***	4.758***
	(1.358)	(1.609)	(1.609)	(1.606)	(1.606)	(1.472)	(1.479)
Random		3.225***		3.245***		3.207***	
		(0.821)		(0.822)		(0.752)	
Complexity			2.230**		2.260*		2.199**
			(1.035)		(1.037)		(0.938)
Requirements			3.650***		3.711***		3.613***
			(1.095)		(1.096)		(1.026)
Welcome			3.787***		3.752***		3.675***
			(1.074)		(1.071)		(0.986)
Panel B. Comparison of policy tree with:							
Random	1.794	1.899		1.759		1.495	
	(1.358)	(1.361)		(1.359)		(1.257)	
Complexity			2.897		2.745		2.559
			(1.539)		(1.538)		(1.432)
Requirements			1.477		1.294		1.144
			(1.503)		(1.499)		(1.409)
Welcome			1.340		1.253		1.083
			(1.528)		(1.527)		(1.415)
Sample	Wave 2	Waves 1 and 2	Waves 1 and 2	Waves 1 and 2	Waves 1 and 2	Waves 1 and 2	Waves 1 and 2
Estimator	OLS	OLS	OLS	PDS lasso	PDS lasso	DDML	DDML
Outcome mean	7.69	7.92	7.92	7.92	7.92	7.92	7.92
Observations	1717	6588	6588	6588	6588	6588	6588

Note: The table reports results from estimating versions of (4) using OLS (columns 1–3), PDS lasso (columns 4 and 5), and DDML (columns 6 and 7). Column 1 only uses data from wave 2; the remaining columns use the full data set. The reference group in column 1 is random treatment allocation; no treatment in columns 2–7. Panel A reports the coefficient estimates. Panel B compares the policy rule using policy trees against always assigning the same treatment to everyone and random treatment allocation. Covariates include the region of nationality, age, gender, years lived in Zurich, and years lived in Switzerland. Standard errors are clustered at building address level.Abbreviations: DDML, double-debiased machine learning; OLS, ordinary least squares; PDS, post-double selection lasso.

*$p < 0.05$, **$p < 0.01$, and ***$p < 0.001$.

4 | CONCLUSION

This paper employs policy trees for assigning eligible immigrants to the information and encouragement treatment that is most likely to address hurdles on their path to citizenship and boost their propensity to naturalize. We evaluate the benefits of this policy rule using a tailored two-phase field experiment. During the exploration phase, we randomly assign eligible immigrants to one of three treatment arms or the control group, based on which we estimate average treatment effects and train the policy tree. We find that despite its simplicity, the optimal policy tree of depth 3 captures more than 85% of the treatment effect heterogeneity (relative to a model-free plug-in rule). Next, we move on to the exploitation phase, in which we assign the subjects that belonged to the control group in the previous phase to either the policy tree or randomly to one of the three treatments. We find that the policy tree outperforms the best-performing individual treatment slightly. While these differences are not statistically significant, it is worth noting that these benefits persist in a context with at most moderate levels of treatment effect heterogeneity and come at little additional costs.

Policy trees possess several advantages that make them particularly suited for policymakers and researchers interested in tailoring treatment assignment to the specific needs of increasingly diverse populations. Policy trees are transparent in terms of which variables guide treatment assignment, they are simple to visualize, and intuitive to communicate even to users of the research who lack statistical training. While using machine learning to personalize treatment assignments raises a host of important ethical and policy questions, we should keep in mind that a one-size-fits-all approach can often exacerbate existing inequalities. For instance, an earlier information letter sent out by the City of Zurich had by far the strongest effects among newly eligible immigrants, which often score higher on multiple integration dimensions compared with more marginalized immigrants who have been residing in the host country for decades without naturalizing

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

(Ward et al., 2019). For all these reasons, we believe that policy trees are a well-suited approach to leverage the potential of tailored treatment assignment in a world where rich background characteristics are increasingly available.

ACKNOWLEDGMENTS

We thank our partners from the City of Zurich for their collaboration and support. We are grateful to Zhengyuan Zhou who has provided helpful feedback and to Teresa Freitas Monteiro for comments. We also thank seminar participants at ETH Zurich, Princeton University, and Stanford University as well as participants at the PolMeth Europe conference. All remaining errors are our own.

OPEN RESEARCH BADGES

This article has been awarded Open Data Badge for making publicly available the digitally-shareable data necessary to reproduce the reported results. Data is available at https://doi.org/10.15456/jae.2024212.1213209091.

DATA AVAILABILITY STATEMENT

The authors provide replication code through the Journal of Applied Econometrics Data Archive. The data are owned by the City of Zurich and the Canton of Zurich. It is not publicly available for confidentiality reasons. Data access requests should be directed to the City of Zurich's population office and the Canton of Zurich's municipal office.

REFERENCES

Ahrens, A., Hansen, C. B., Schaffer, M. E., & Wiemann, T. (2024). Double machine learning and model averaging. https://arxiv.org/abs/2401.01645

Assunção, J., McMillan, R., Murphy, J., & Souza-Rodrigues, E. (2022). Optimal environmental targeting in the amazon rainforest. *The Review of Economic Studies*, *90*, 1608–1641.

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

Athey, S., & Wager, S. (2021). Policy learning with observational data. *Econometrica*, *89*(1), 133–161.

Bauböck, R., Ersbøll, E., Groenendijk, K., & Waldrauch, H. (2006). *Acquisition and loss of nationality: Comparative analyses - Policies and trends in 15 European countries*. Amsterdam University Press.

Belloni, A., Chernozhukov, V., & Hansen, C. (2014). 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., & 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.

Bhargava, S., & Manoli, D. (2015). Psychological frictions and the incomplete take-up of social benefits: Evidence from an irs field experiment. *American Economic Review*, *105*(11), 3489–3529.

Bhattacharya, D., & Dupas, P. (2012). Inferring welfare maximizing treatment assignment under budget constraints. *Journal of Econometrics*, *167*(1), 168–196.

Blizzard, B., & Batalova, J. (2019). Naturalization trends in the United States States. Tech. rep. Migration Policy Institute. https://www.migrationpolicy.org/article/naturalizationtrends-united-states-2017

Bloemraad, I. (2002). The North American naturalization gap: An institutional approach to citizenship acquisition in the United States and Canada. *International Migration Review*, *36*(1), 193–228.

Bloemraad, I., Korteweg, A., & Yurdakul, G. (2008). Citizenship and immigration: Multiculturalism, assimilation, and challenges to the nation-state. *Annual Review of Sociology*, *34*(1), 153–179.

Cagala, T., Glogowsky, U., Rincke, J., & Strittmatter, A. (2021). Optimal targeting in fundraising: A machine-learning approach. SSRN Electronic Journal.

Caria, S., Kasy, M., Quinn, S., Shami, S., Teytelboym, A., et al. (2020). An adaptive targeted field experiment: Job search assistance for refugees in Jordan Center for Economic Studies & Ifo Institute. http://doi.org/10.2139/ssrn.3689456

Chen, W., Hribar, P., & Melessa, S. (2023). Standard error biases when using generated regressors in accounting research. *Journal of Accounting Research*, *61*(2), 531–569.

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., Demirer, M., Duflo, E., & Fernández-Val, I. (2018). Generic machine learning inference on heterogeneous treatment effects in randomized experiments, with an application to immunization in india. (*24678*), National Bureau of Economic Research. http://www.nber.org/papers/w24678

Dancygier, R. M. (2010). *Immigration and conflict in Europe*. New York: Cambridge University Press English.

Davis, J. M. V., & Heller, S. B. (2017). Using causal forests to predict treatment heterogeneity: An application to summer jobs. *American Economic Review*, *107*(5), 546–550.

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

European Commission. (2021). Laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts. Tech. rep. https://eurlex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0206&from=EN

Felfe, C., Kocher, M. G., Rainer, H., Saurer, J., & Siedler, T. (2021). More opportunity, more cooperation? The behavioral effects of birthright citizenship on immigrant youth. *Journal of Public Economics*, *200*, 104448.

Finkelstein, A., & Notowidigdo, M. J. (2019). Take-up and targeting: Experimental evidence from SNAP. *The Quarterly Journal of Economics*, *134*(3), 1505–1556.

Frölich, M. (2008). Statistical treatment choice: An application to active labor market programs. *Journal of the American Statistical Association*, *103*(482), 547–558 en.

Gabaix, X. (2019). Behavioral inattention, *Handbook of behavioral economics: Applications and foundations 1* (Vol. *2*, pp. 261–343). Elsevier. https://linkinghub.elsevier.com/retrieve/pii/S2352239918300216

Gathmann, C., & Garbers, J. (2023). Citizenship and integration. *Labour Economics*, *82*, 102343.

Gathmann, C., & Keller, N. (2018). Access to citizenship and the economic assimilation of immigrants. *The Economic Journal*, *128*(616), 3141–3181.

Gerarden, T., & Yang, M. (2022). Using targeting to optimize program design: Evidence from an Energy Conservation Experiment. *Journal of the Association of Environmental and Resource Economists*, *10*(3), 687–716.

Goldin, J., Homonoff, T., Javaid, R., & Schafer, B. (2022). Tax filing and take-up: Experimental evidence on tax preparation outreach and benefit claiming. *Journal of Public Economics*, *206*, 104550.

Gonzalez-Barrera, A., Lopez, M. H., Passel, J. S., & Taylor, P. (2013). The path not taken: Pew Hispanic Center, February. https://assets.pewresearch.org/wp- content/uploads/sites/7/2013/02/Naturalizations_ Jan_2013_FINAL.pdf

Goodman, S. W. (2014). *Immigration and membership politics in western Europe*. Cambridge University Press.

Govind, Y. (2021). *Is naturalization a passport for better labor market integration?: Evidence from a quasi-experimental setting*. Paris School of Economics.

Haaland, I., Roth, C., & Wohlfart, J. (2023). Designing information provision experiments. *Journal of Economic Literature*, *61*(1), 3–40.

Hainmueller, J., & Hangartner, D. (2013). Who gets a Swiss passport? A natural experiment in immigrant discrimination. *American Political Science Review*, *107*(01), 159–187.

Hainmueller, J., Hangartner, D., & Pietrantuono, G. (2015). Naturalization fosters the long-term political integration of immigrants. *Proceedings of the National Academy of Sciences*, *112*(41), 12651–12656.

Hainmueller, J., Hangartner, D., & Pietrantuono, G. (2017). Catalyst or crown: Does naturalization promote the long-term social integration of immigrants? *American Political Science Review*, *111*(2), 256–276.

Hainmueller, J., Hangartner, D., & Ward, D. (2019). The effect of citizenship on the long-term earnings of marginalized immigrants: Quasi-experimental evidence from Switzerland. *Science Advances*, *5*(12), eaay1610.

Handel, B., & Schwartzstein, J. (2018). Frictions or mental gaps: What's behind the information we (don't) use and when do we care? *Journal of Economic Perspectives*, *32*(1), 155–178.

Hangartner, D., Ward, D., Stampi-Bombelli, A., & Kurer, S. (2023). How can citizenship campaigns overcome hurdles to immigrant naturalization? Experimental evidence from two randomized control trials. Unpublished manuscript.

Hirano, K., & Porter, J. R. (2009). Asymptotics for statistical treatment rules. *Econometrica*, *77*(5).

Hirano, K., & Porter, J. R. (2020). Asymptotic analysis of statistical decision rules in econometrics. In Durlauf, S. N., Hansen, L. P., Heckman, J. J., & Matzkin, R. L. (Eds.), *Handbook of econometrics*. Handbook of econometrics, volume 7A, (Vol. 7, pp. 283–354). Elsevier. https://www.sciencedirect.com/science/article/pii/ S1573441220300040

Hotard, M., Lawrence, D., Laitin, D. D., & Hainmueller, J. (2019). A low-cost information nudge increases citizenship application rates among low-income immigrants. *Nature human behaviour*, *3*, 678–683.

Huddleston, T. (2013). The naturalisation procedure: Measuring the ordinary obstacles and opportunities for immigrants to become citizens.

Ida, T., Ishihara, T., Ito, K., Kido, D., Kitagawa, T., Sakaguchi, S., & Sasaki, S. (2022). Choosing who chooses: Selection-driven targeting in energy rebate programs. (*30469*) National Bureau of Economic Research. http://www.nber.org/papers/w30469

Imbens, G. W. (2000). The role of the propensity score in estimating dose-response functions. *Biometrika*, *87*(3), 706–710. Publisher: [Oxford University Press, Biometrika Trust].

Imbens, G. (2004). Nonparametric estimation of average treatment effects under exogeneity: A review. Review of Economics and Statistics.

Imbens, G. W., & Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

Jacob, D. (2021). CATE meets ML. *Digital Finance*, *3*(2), 99–148.

Keller, N., Gathmann, C., & Monscheuer, O. (2015). Citizenship and the social integration of immigrants: Evidence from Germany's immigration reforms.

Kitagawa, T., & Tetenov, A. (2018). Who should be treated? Empirical welfare maximization methods for treatment choice. *Econometrica*, *86*(2), 591–616.

Kitagawa, T., & Wang, G. (2023). Who should get vaccinated? Individualized allocation of vaccines over SIR network. *Journal of Econometrics*, *232*(1), 109–131.

Knaus, M. C., Lechner, M., & Strittmatter, A. (2021). Machine learning estimation of heterogeneous causal effects: Empirical Monte Carlo evidence. *The Econometrics Journal*, *24*(1), 134–161.

Knaus, M. C., Lechner, M., & Strittmatter, A. (2022). Heterogeneous employment effects of job search programs. *Journal of Human Resources*, *57*(2), 597–636.

Knittel, C. R., & Stolper, S. (2021). Machine learning about treatment effect heterogeneity: The case of household energy use. *AEA Papers and Proceedings*, *111*, 440–444.

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, *116*(10), 4156–4165.
Lechner, M., & Smith, J. (2007). What is the value added by caseworkers? *Labour Economics*, *14*(2), 135–151.
Maćkowiak, B., Matejka, F., & Wiederholt, M. (2023). Rational inattention: A review. *Journal of Economic Literature*, *61*(1), 226–273.
Manski, C. F. (2004). Statistical treatment rules for heterogeneous populations. *Econometrica*, *72*(4), 1221–1246.
Manski, C. F. (2007). Minimax-regret treatment choice with missing outcome data. *Journal of Econometrics*, *139*(1), 105–115.
Manski, C. F. (2021). Econometrics for decision making: Building foundations sketched by Haavelmo and Wald. *Econometrica*, *89*(6), 2827–2853.
Mazzolari, F. (2009). Dual citizenship rights: Do they make more and richer citizens? *Demography*, *46*(1), 169–191.
Murphy, K. M., & Topel, R. H. (1985). Estimation and inference in two-step econometric models. *Journal of Business & Economic Statistics*, *3*(4), 370–379.
National Academies of Sciences, et al. (2016). *The integration of immigrants into American society*. National Academies Press.
OECD. (2011). *Naturalisation: A passport for the better integration of immigrants?* OECD Publishing.
Pagan, A. (1984). Econometric issues in the analysis of regressions with generated regressors. *International Economic Review*, *25*(1), 221–247. Publisher: [Economics Department of the University of Pennsylvania, Wiley, Institute of Social and Economic Research, Osaka University].
Pastor, M., & Scoggins, J. (2012). The economic benefits of naturalization for immigrants and the economy. Center for the Study of Immigrant Integration's, University of Southern California.
Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*(1), 41–55.
Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, *66*, 688–701. Place: US Publisher: American Psychological Association.
Savage, L. J. (1951). The theory of statistical decision. *Journal of the American Statistical Association*, *46*(253), 55–67. Publisher: American Statistical Association, Taylor & Francis, Ltd.
Stoye, J. (2009). Minimax regret treatment choice with finite samples. *Journal of Econometrics*, *151*(1), 70–81.
Stoye, J. (2012). Minimax regret treatment choice with covariates or with limited validity of experiments. *Journal of Econometrics*, *166*(1), 138–156.
Sverdrup, E., Kanodia, A., Zhou, Z., Athey, S., & Wager, S. (2022). policytree: Policy learning via doubly robust empirical welfare maximization over trees. R package version 1.2.1. https://CRAN.R-project.org/ package=policytree
Swaminathan, A., & Joachims, T. (2015). Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, *16*(52), 1731–1755.
The White House (2022). *Blueprint for an AI bill of rights. Making automated systems work for the American people*. White House Office of Science and Technology Policy. https://www.whitehouse.gov/ostp/ ai-bill-of-rights
Tibshirani, J., Athey, S., Sverdrup, E., & Wager, S. (2022). GRF: Generalized random forests. R package version 2.2.0. https://CRAN.R-project.org/package=grf
Vernby, K., & Dancygier, R. (2019). Can immigrants counteract employer discrimination? A factorial field experiment reveals the immutability of ethnic hierarchies. *PloS one*, *14*(7), e0218044.
Vink, M., Tegunimataka, A., Peters, F., & Bevelander, P. (2021). Long-term heterogeneity in immigrant naturalisation: The conditional relevance of civic integration and dual citizenship. *European Sociological Review*, *37*, 751–765.
Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*(523), 1228–1242. Publisher: Taylor & Francis.
Wald, A. (1950). *Statistical decision functions*. Wiley.
Ward, D., Pianzola, J., & Hangartner, D. (2019). Large-scale information campaigns can increase naturalization rates. Unpublished Manuscript.
Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data*. MIT Press.
Zhou, Z., Athey, S., & Wager, S. (2022). Offline multi-action policy learning: Generalization and optimization. *Operations Research*, *71*(1), 148–183.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of the article.

How to cite this article: Ahrens, A., Stampi-Bombelli, A., Kurer, S., Hangartner, D. (2024). Optimal multi-action treatment allocation: A two-phase field experiment to boost immigrant naturalization. *Journal of Applied Econometrics*, 1–17. https://doi.org/10.1002/jae.3092

10991255, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/jae.3092 by Eth Zurich, Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License