On the Computational Stability of Cross-Fitted Double Machine Learning

Anonymous authors
Paper under double-blind review

Abstract

Double machine learning (DML) combines orthogonal score construction with cross-fitting to enable semiparametric inference using flexible machine learning (ML) nuisance estimators. While the statistical properties of DML have been studied extensively, comparatively little attention has been devoted to the computational variability induced by randomized sample splitting itself. In practical implementations, repeated executions of the same DML procedure on a fixed dataset may produce different parameter estimates solely because randomized fold assignments vary across runs. This paper investigates the finite-sample computational stability of cross-fitted DML estimators under repeated randomized sample splitting. We introduce a simple split-instability metric that quantifies estimator variability across independently generated cross-fitting realizations and study its empirical behavior through controlled simulation experiments. The experiments demonstrate that randomized cross-fitting induces non-negligible finite-sample variability, even when the underlying dataset and nuisance learners remain fixed. Repeated cross-fitting substantially stabilizes DML estimates and empirically exhibits behavior closely consistent with inverse square-root variance reduction under repeated averaging. Additional experiments show that instability decreases with increasing sample size, increases in highly correlated high-dimensional regimes and depends nontrivially on nuisance learner choice and cross-fitting configuration. Across all experiments, repeated cross-fitting primarily improves estimator performance through variance reduction rather than bias reduction. Overall, the results highlight the importance of computational reproducibility diagnostics in ML-assisted inference and suggest that repeated cross-fitting provides a simple stabilization mechanism for practical DML workflows.

1 Introduction

Double machine learning (DML) has emerged as a widely used framework for semiparametric and causal inference in high-dimensional settings (Chernozhukov et al., 2018). By combining orthogonal score construction with modern machine learning (ML) nuisance estimators, DML enables inference on low-dimensional target parameters, while controlling for complex nuisance components. The framework has become increasingly important in econometrics, statistics and ML due to its flexibility, scalability and favorable asymptotic properties.

A central component of DML is cross-fitting, in which the observed sample is partitioned into training and evaluation folds. Nuisance functions are estimated on training subsets and subsequently evaluated on held-out folds, in order to reduce overfitting bias and improve robustness (Chernozhukov et al., 2018; Newey & Robins, 2018). Cross-fitting has become a standard ingredient in orthogonalized estimation procedures and related debiased ML frameworks. Despite extensive theoretical work on orthogonality, asymptotic efficiency and semiparametric inference, relatively little focus has been placed on the computational variability induced by randomized sample splitting itself. In practice, repeated executions of the same DML pipeline on a fixed dataset can produce different parameter estimates purely due to randomized fold assignments change across runs. Although such variability is often implicitly treated as negligible or hidden through fixed random seeds, it may remain practically observable in moderate sample sizes and highly correlated high-dimensional settings.

Understanding this phenomenon is important for several reasons. First, many empirical studies report results obtained from a single randomized cross-fitting realization, potentially introducing unreported algorithmic variability into the final estimate. Second, randomized computational procedures are increasingly common throughout ML and modern statistical inference, making computational reproducibility an important methodological consideration. Third, split-induced instability may interact nontrivially with nuisance learner complexity, regularization and finite-sample estimation error.

This paper investigates the finite-sample computational stability of cross-fitted DML estimators under repeated randomized sample splitting. Rather than focusing primarily on asymptotic theory, we study the empirical behavior of DML estimators across independently generated fold assignments, while holding the underlying dataset fixed. We introduce a simple split-instability metric that quantifies estimator dispersion induced solely by cross-fitting randomness and investigate its behavior under varying sample sizes, repeated cross-fitting repetitions, nuisance learners, fold configurations and high-dimensional correlation structures.

Our experiments show that randomized cross-fitting introduces non-negligible finite-sample variability, even when the dataset and model specification remain unchanged. We further show that repeated cross-fitting substantially improves the stability of DML estimates and follows a pattern closely aligned with inverse square-root variance reduction under repeated averaging. Additional experiments indicate that instability decreases with increasing sample size, becomes amplified in highly correlated high-dimensional settings and depends nontrivially on nuisance learner choice and cross-fitting configuration.

The experiments further suggest that repeated cross-fitting primarily improves estimator performance through reduction of split-induced variance rather than reduction of estimator bias. Across all simulated experimental settings, averaging across independently generated fold assignments consistently improves estimator stability and reproducibility. The results additionally reveal a practical tradeoff between estimator stabilization and computational cost as the number of repeated cross-fitting repetitions increases.

The contributions of this work are summarized as follows:

1. We introduce a simple computational instability metric for quantifying estimator variability induced by randomized cross-fitting.

2. We demonstrate empirically that cross-fitted DML estimators exhibit measurable finite-sample instability across independently generated sample splits.

3. We show that repeated cross-fitting substantially stabilizes DML estimation and empirically exhibits approximately inverse square-root variance reduction behavior.

4. We investigate how instability varies with sample size, nuisance learner choice, cross-fitting configuration and high-dimensional correlation structure.

5. We provide extensive simulation evidence demonstrating the practical importance of computational reproducibility diagnostics in ML-assisted inference.

The remainder of the paper is organized as follows. Section 2 reviews the DML framework and randomized cross-fitting procedures. Section 3 introduces the proposed split-instability framework and repeated cross-fitting methodology. Section 4 describes the experimental design and datasets. Section 5 presents the empirical results. Section 6 discusses the implications of the findings. Section 7 outlines limitations and directions for future work and Section 8 concludes.

2 Background

We consider the partially linear regression model

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

where Y denotes the outcome variable, D denotes the treatment variable, $X \in \mathbb{R}^p$ is a potentially high-dimensional covariate vector, g_0 is an unknown nuisance function and θ_0 is the low-dimensional target parameter of interest. The random error term ε satisfies suitable moment conditions together with the orthogonality condition

$$\mathbb{E}[\varepsilon \mid X, D] = 0 \ . \tag{2}$$

DML estimates θ_0 by combining orthogonal score construction with flexible ML-based nuisance estimation (Chernozhukov et al., 2018). The framework is designed to permit valid inference on low-dimensional target parameters, even when nuisance components are estimated using high-dimensional or nonparametric ML procedures. For the partially linear regression setting, DML typically estimates nuisance functions of the form

$$m_0(X) = \mathbb{E}[D \mid X] \tag{3}$$

and

$$\ell_0(X) = \mathbb{E}[Y \mid X] \ , \tag{4}$$

using ML procedures such as regularized regression, random forests, boosting methods or neural networks (Breiman, 2001; Friedman, 2001; Bishop, 2006; Hastie et al., 2009). The target parameter is subsequently estimated using orthogonalized residuals constructed from these nuisance estimates. A commonly used orthogonal score for the partially linear model is

$$\psi(W; \theta, \eta) = (D - m(X)) \left(Y - \ell(X) - \theta(D - m(X))\right) \ , \tag{5}$$

where $W = (Y, D, X)$ and $\eta = (m, \ell)$ denotes the nuisance components. The orthogonality property implies that first-order perturbations in nuisance estimation affect the target parameter estimate only at higher order, thereby reducing sensitivity to regularization bias.

The resulting DML estimator is obtained by solving the empirical orthogonal score equation using cross-fitted nuisance estimates. In the partially linear regression setting, the estimator can be written explicitly as

$$\hat{\theta} = \frac{\sum_{i=1}^{n} \left(D_i - \hat{m}_{-k(i)}(X_i)\right) \left(Y_i - \hat{\ell}_{-k(i)}(X_i)\right)}{\sum_{i=1}^{n} \left(D_i - \hat{m}_{-k(i)}(X_i)\right)^2} \ , \tag{6}$$

where $\hat{m}_{-k(i)}$ and $\hat{\ell}_{-k(i)}$ denote nuisance estimators trained using observations outside the fold containing observation i. The notation emphasizes that nuisance estimation and score evaluation are separated through cross-fitting.

A key component of DML is cross-fitting. Let the observed sample be partitioned into K folds. For each fold, nuisance functions are estimated on the complementary training folds and evaluated on the held-out fold. Orthogonal score contributions are then aggregated across all folds to obtain the final estimator (Chernozhukov et al., 2018; Newey & Robins, 2018). More formally, if $\mathcal{I}_k$ denotes the index set corresponding to the k^{th} fold, nuisance estimators are trained using observations outside $\mathcal{I}_k$ and evaluated using observations inside $\mathcal{I}_k$. Cross-fitting reduces dependence between nuisance estimation and score evaluation, thereby improving finite-sample robustness.

Although cross-fitting improves statistical performance, it also introduces algorithmic randomness through randomized fold assignment. Consequently, repeated executions of the same DML procedure on the same dataset may produce different parameter estimates solely because different sample partitions are generated

during cross-fitting. In large samples, this variability is often implicitly assumed to be negligible relative to classical sampling uncertainty. However, in moderate sample sizes or highly correlated high-dimensional settings, split-induced fluctuations may remain practically observable and may affect computational reproducibility. While asymptotic analyses typically average over the randomness induced by sample splitting, finite-sample implementations may still exhibit non-negligible sensitivity to particular fold realizations. Consequently, understanding the empirical magnitude of this computational variability is important for assessing estimator robustness and reproducibility in practice.

The present work focuses specifically on this source of variability. Rather than studying asymptotic efficiency or orthogonality properties, we investigate the computational stability of DML estimators under repeated randomized cross-fitting and study whether repeated averaging across independently generated fold assignments can provide a practical stabilization mechanism. The resulting perspective connects DML with broader themes in computational statistics and ML concerning algorithmic randomness, reproducibility and stability under randomized procedures (Efron & Tibshirani, 1994; Bousquet & Elisseeff, 2002; Xu & Mannor, 2012). Figure 1 summarizes the computational workflow underlying the proposed instability framework and illustrates how repeated randomized cross-fitting generates split-specific estimators, whose variability is subsequently stabilized through repeated averaging.

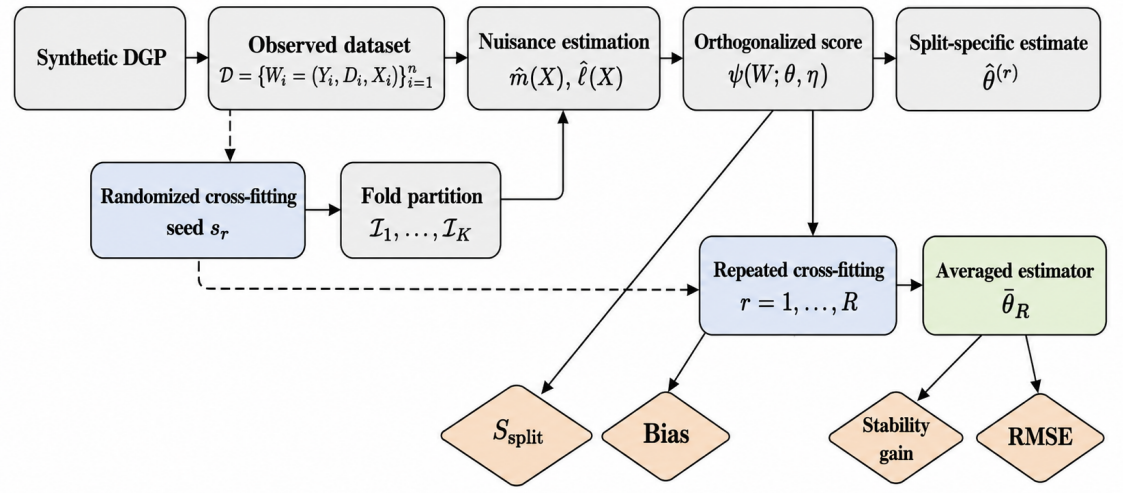


Figure 1: Overview of the proposed computational stability framework for cross-fitted DML. Randomized cross-fitting generates split-specific DML estimates $\hat{\theta}^{(r)}$ through fold-dependent nuisance estimation and orthogonalized score evaluation. Repeated averaging across independently generated fold assignments yields the stabilized estimator $\bar{\theta}_R$. Split-induced variability is quantified using the instability metric S_{split}.

3 Split Instability and Repeated Cross-Fitting

To quantify the computational variability induced by randomized cross-fitting, we introduce an instability metric based on repeated executions of the DML procedure on a fixed dataset. As illustrated in Figure 1, independently randomized cross-fitting realizations generate split-specific DML estimates, whose dispersion forms the basis of the proposed instability diagnostics.

Formally, let

$$\hat{\theta}^{(r)} = \mathcal{A}(\mathcal{D}, S_r) \ , \tag{7}$$

where $\mathcal{D}$ denotes the fixed observed dataset, S_r denotes the randomized cross-fitting assignment generated at repetition r and $\mathcal{A}$ denotes the complete DML estimation procedure. This representation makes explicit

that repeated executions differ only through randomized fold assignment, while the underlying dataset and nuisance learners remain unchanged.

Let $\hat{\theta}^{(1)}, \hat{\theta}^{(2)}, \ldots, \hat{\theta}^{(R)}$ denote DML estimates obtained using R independently generated randomized cross-fitting assignments, while keeping the observed dataset fixed. We define the split-instability measure as

$$S_{\text{split}} = \text{Std}\left(\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(R)}\right) \ , \tag{8}$$

where $\text{Std}(\cdot)$ denotes the empirical standard deviation. The quantity S_{split} measures the sensitivity of the estimator to randomized fold assignment. Small values indicate that the DML estimator is computationally stable with respect to cross-fitting randomness, while larger values indicate increased algorithmic variability. The proposed instability measure is analogous to a conditional Monte Carlo standard deviation computed over randomized cross-fitting realizations, while conditioning on the observed dataset.

Importantly, the proposed metric isolates only variability arising from randomized sample splitting. Across repetitions, the observed dataset, nuisance learners and model specification remain fixed. Consequently, the resulting estimator dispersion reflects solely the computational randomness introduced by cross-fitting. From a computational perspective, the instability measure can be interpreted as a form of algorithmic sensitivity. Different randomized fold assignments produce slightly different nuisance estimates and orthogonal score evaluations, leading to corresponding fluctuations in the final DML estimate. The resulting estimator distribution therefore characterizes the practical reproducibility of the DML procedure under repeated randomized execution.

The proposed framework connects DML with broader notions of algorithmic stability and resampling variability that appear throughout ML and computational statistics (Bousquet & Elisseeff, 2002; Arlot & Celisse, 2010; Xu & Mannor, 2012). However, unlike bootstrap procedures, the present framework holds the observed dataset fixed and isolates only variability induced by randomized cross-fitting itself. This perspective allows split-induced estimator variability to be studied as a distinct computational component of uncertainty in ML-assisted inference procedures.

A central perspective of the present work is that repeated cross-fitting can be interpreted as a computational variance reduction mechanism operating over independently randomized cross-fitting realizations. We define the repeated cross-fit averaged estimator

$$\bar{\theta}_R = \frac{1}{R} \sum_{r=1}^{R} \hat{\theta}^{(r)} \ , \tag{9}$$

where each $\hat{\theta}^{(r)}$ corresponds to an independently randomized cross-fitting realization. To study the stabilization induced by repeated averaging, we define the averaged-estimator instability measure

$$S_{\text{avg}}(R) = \text{Std}\left(\bar{\theta}_R^{(1)}, \ldots, \bar{\theta}_R^{(M)}\right) \ , \tag{10}$$

where M denotes the number of independently constructed repeated cross-fit averaged estimators used to estimate instability empirically and each $\bar{\theta}_R^{(m)}$ is computed using R independently randomized fold assignments.

If repeated cross-fitted estimators exhibit sufficiently weak dependence across independently randomized fold assignments and possess finite variance, averaging across repetitions is expected to reduce estimator variability approximately according to

$$\text{Var}(\bar{\theta}_R) \propto R^{-1} \ , \tag{11}$$

suggesting stabilization behavior of the form

$$S_{\text{avg}}(R) \propto R^{-1/2} \ . \tag{12}$$

This behavior is analogous to classical variance reduction phenomena encountered in Monte Carlo simulation and ensemble averaging. The statement should be interpreted as an informal variance reduction heuristic rather than a formal asymptotic theorem, since repeated cross-fitted estimators generated from the same observed dataset are generally dependent through shared observations and nuisance estimation structure. Nevertheless, the empirical experiments reported later exhibit behavior closely consistent with inverse square-root stabilization.

The distinction between estimator bias and split-induced variance is also important. Repeated cross-fitting primarily affects estimator variability through averaging across independently generated fold assignments. In contrast, systematic finite-sample bias arising from nuisance misspecification or regularization may remain largely unchanged. Consequently, the experiments separately evaluate instability, bias and root-mean-square error (RMSE).

The proposed instability diagnostics are computationally lightweight and straightforward to implement. Since repeated cross-fitting executions are independently randomized conditional on the observed dataset, the procedure can additionally be parallelized efficiently across computational cores or distributed systems. However, increasing the number of repeated cross-fitting repetitions also increases computational cost, thereby introducing a practical tradeoff between estimator stabilization and runtime. Overall, the proposed framework provides a simple and computationally practical methodology for studying reproducibility and algorithmic uncertainty in modern ML-assisted inference procedures, while simultaneously serving as a practical reproducibility diagnostic for randomized cross-fitting pipelines.

In addition, the proposed instability framework motivates a conceptual decomposition between classical sampling variability and computational variability induced by randomized cross-fitting. Informally, the overall estimator variance may be decomposed as

$$\text{Var}(\hat{\theta}) = \underbrace{\text{Var}_{\mathcal{D}}\left(\mathbb{E}_S[\hat{\theta} \mid \mathcal{D}]\right)}_{\text{sampling variability}} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\text{Var}_S(\hat{\theta} \mid \mathcal{D})\right]}_{\text{split-induced variability}} \ , \tag{13}$$

where the outer expectation is taken over repeated datasets, while the inner variance is taken over randomized cross-fitting realizations conditional on a fixed dataset. The proposed instability metric specifically targets the second component, thereby isolating computational variability induced solely by randomized sample splitting.

4 Experimental Setup

We study the computational stability of cross-fitted DML estimators using controlled synthetic experiments. The simulation framework is intentionally designed to isolate variability induced solely by randomized cross-fitting, while maintaining full reproducibility and modest computational cost. Synthetic experiments are particularly useful in the present setting because the underlying data-generating mechanism remains fixed, while randomized fold assignments vary across repeated executions. This controlled setup therefore isolates the computational effect of cross-fitting randomness independently of additional sampling heterogeneity.

4.1 Synthetic Data Generation

Covariates are generated from a multivariate Gaussian distribution

$$X \sim \mathcal{N}(0, \Sigma) \ , \tag{14}$$

where the covariance matrix follows the Toeplitz structure

$$\Sigma_{ij} = \rho^{|i-j|} \ , \tag{15}$$

with correlation parameter $\rho \in (0, 1)$. The synthetic data-generating process follows the partially linear regression framework introduced in Section 2. The treatment and outcome variables are generated according to

$$D = X\beta_D + v \ , \tag{16}$$
$$Y = D\theta_0 + g_0(X) + \varepsilon \ , \tag{17}$$

where

$$v, \varepsilon \sim \mathcal{N}(0, 1) \tag{18}$$

are independent Gaussian noise variables. The baseline experiments use a linear nuisance specification

$$g_0(X) = X\beta_Y \ , \tag{19}$$

while additional robustness experiments consider nonlinear nuisance structure of the form

$$g_0(X) = X\beta_Y + 0.5 \sin(X_1) + 0.5X_2^2 - 0.5X_3X_4 \ . \tag{20}$$

The coefficient vectors β_D and β_Y contain sparse low-dimensional signal components, while the remaining coefficients are set to zero. This design reflects common sparse high-dimensional settings frequently studied in statistical learning and causal inference (Vapnik, 1998; Bühlmann & van de Geer, 2011). Unless otherwise stated, experiments use a baseline moderate-dimensional partially linear regression configuration. Full simulation parameters are provided in Appendix A.

To investigate instability amplification in more difficult regimes, we additionally consider a highly correlated high-dimensional configuration satisfying

$$(n, p, \rho) = (100, 500, 0.95) \ , \tag{21}$$

where n denotes the sample size, p denotes the covariate dimension and ρ denotes the correlation parameter governing the Toeplitz covariance structure. This setting substantially increases nuisance estimation difficulty and empirically amplifies split-induced computational variability.

4.2 Nuisance Learners

The experiments consider ridge, lasso and random forest nuisance learners, in order to study instability behavior across both linear and nonlinear estimation procedures. These learners were selected because they represent widely used linear and nonlinear ML procedures with different regularization and prediction characteristics. The random forest learner uses fixed internal random seeds, in order to isolate instability arising specifically from randomized cross-fitting rather than from internal learner randomness.

4.3 DML Implementation

All DML procedures are implemented using the `DoubleML` Python package (Bach et al., 2022; Chernozhukov et al., 2022). The implementation follows the partially linear regression framework with externally controlled randomized fold assignment. To isolate computational variability induced by sample splitting, all repeated executions use independently generated cross-fitting partitions, while holding the underlying dataset fixed. Additional implementation and reproducibility details are provided in Appendix B.

Table 1: Effect of sample size on split instability

Sample Size (n)	Mean Instability	Monte Carlo Std. Error
100	0.0632	0.0055
150	0.0475	0.0023
300	0.0274	0.0011
500	0.0139	0.0009

4.4 Experimental Configurations

The experiments investigate several aspects of computational instability in cross-fitted DML estimation:

1. the relationship between sample size and split instability
2. the stabilization behavior induced by repeated cross-fitting
3. instability amplification in highly correlated high-dimensional settings
4. robustness across different nuisance learners
5. sensitivity to the number of cross-fitting folds
6. interactions between fold configuration and repeated cross-fitting averaging
7. comparisons between no-crossfit, single-crossfit and repeated-crossfit estimation procedures.

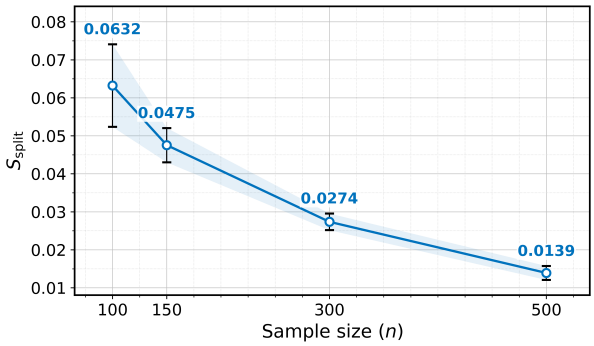
Baseline simulation parameters are reported in Appendix A. Each experimental configuration is evaluated across multiple independently generated datasets. Within each dataset, repeated randomized fold assignments are generated independently. Reported instability measures therefore average across both repeated cross-fitting realizations and multiple simulation datasets. The primary quantities reported throughout the experiments are the instability measures introduced in Section 3. In addition to instability, we also report empirical bias, RMSE and computational runtime, in order to distinguish variance reduction effects from systematic finite-sample bias behavior and computational cost. The experiments therefore evaluate both statistical and computational aspects of repeated cross-fitting stabilization.

5 Results

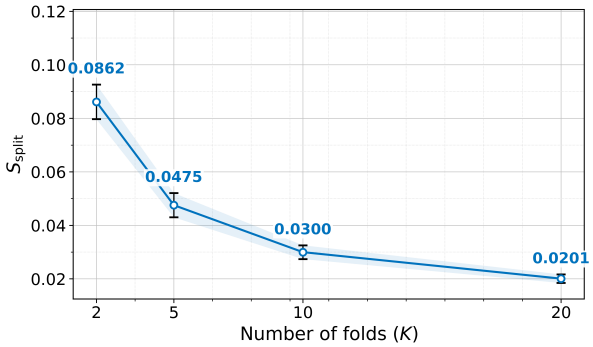
5.1 Sample Size and Split Instability

We first investigate the relationship between sample size and split-induced estimator variability under the baseline linear nuisance specification. Figure 2a presents the empirical split-instability measure as a function of sample size. The overall trend indicates decreasing estimator instability as the number of observations increases. Larger datasets consistently produce more stable DML estimates and lower sensitivity to cross-fitting randomness. Intuitively, increasing the sample size improves nuisance estimation accuracy and reduces the dependence of orthogonal score evaluation on individual sample splits. Consequently, the relative contribution of randomized cross-fitting variability diminishes as more observations become available.

The decrease in empirical instability is particularly noticeable between $n = 100$ and $n = 500$, where the mean instability decreases from approximately 0.063 to 0.014. This reduction suggests that split-induced variability behaves primarily as a finite-sample computational phenomenon. Figure 2a illustrates the stabilization trend, while Table 1 reports the corresponding numerical values. The instability values decrease monotonically across all considered sample sizes, providing consistent evidence that larger datasets reduce the impact of cross-fitting randomness.



(a) Split instability decreases with increasing sample size under the baseline linear nuisance specification.



(b) Split instability decreases as the number of cross-fitting folds increases.

Figure 2: Split-instability diagnostics under varying sample sizes and fold configurations. Error bars correspond to 95% Monte Carlo confidence intervals computed across independently generated datasets.

5.2 Repeated Cross-Fitting Stabilization

We next investigate the effect of repeated cross-fitting on estimator instability. Figure 3a presents the empirical behavior of the averaged-estimator instability measure $S_{\text{avg}}(R)$ as the number of repeated cross-fitting repetitions increases. The results demonstrate substantial stabilization as the number of repetitions increases. In the reported experiments, repeated cross-fitting reduced estimator instability by approximately a factor of 7 between $R = 1$ and $R = 50$. Empirically, the observed stabilization behavior is closely consistent with the variance reduction heuristic developed in Section 3, with instability decreasing systematically under repeated cross-fit averaging. The reduction in instability is already substantial for relatively small repetition counts. Increasing the number of repetitions from $R = 1$ to $R = 10$ reduces instability by roughly a factor of 3, while additional repetitions continue to provide diminishing but consistent stabilization gains.

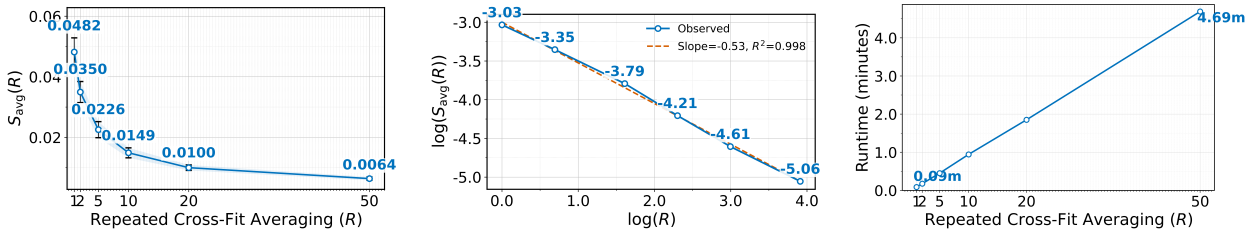
Table 2 indicates a nearly monotone reduction in instability and RMSE as the number of repeated cross-fitting repetitions increases. In contrast, the empirical bias remains approximately constant across repetitions. These findings suggest that repeated cross-fitting primarily improves estimator performance through reduction of split-induced variance rather than reduction of estimator bias.

Under approximate inverse square-root stabilization behavior, the averaged-estimator instability is expected to satisfy the approximate log-linear relationship

$$\log S_{\text{avg}}(R) = \alpha - \frac{1}{2} \log R \ , \tag{22}$$

for some constant α. Consequently, inverse square-root stabilization behavior appears approximately linear on the log-log scale. To further investigate the stabilization behavior, Figure 3b presents a log-log representation of the relationship between repeated cross-fitting repetitions and estimator instability. The estimated log-log slope is approximately -0.53, remarkably close to the inverse square-root stabilization behavior expected under repeated averaging.

The experiments additionally reveal a practical computational tradeoff. Increasing the number of repeated cross-fitting repetitions improves estimator stability, but also increases runtime approximately linearly in the number of repetitions, as shown in Figure 3c. Nevertheless, substantial stabilization is already achieved for moderate repetition counts such as $R \in \{5, 10, 20\}$, suggesting that repeated cross-fitting provides favorable stability improvements at relatively manageable computational cost.



(a) Repeated cross-fitting substantially reduces split-induced estimator instability. Error bars correspond to 95% Monte Carlo confidence intervals computed across independently generated datasets.

(b) Log-log relationship between repeated cross-fitting repetitions and estimator instability. The observed slope is approximately consistent with inverse-square-root stabilization behavior under repeated averaging.

(c) Runtime scaling as the number of repeated cross-fitting repetitions increases. Computational cost grows approximately linearly with the number of repetitions across the reported experimental settings.

Figure 3: Effects of repeated cross-fitting on estimator stability and computational cost

Table 2: Repeated cross-fitting stabilization results. Reported values are means across simulated datasets with standard errors in parentheses.

Repetitions (R)	Instability	Absolute Bias	RMSE	Runtime (s)
1	0.0482 (0.0024)	0.0874 (0.0246)	0.1069 (0.0208)	5.29
2	0.0350 (0.0018)	0.0873 (0.0241)	0.0986 (0.0219)	10.60
5	0.0226 (0.0013)	0.0885 (0.0252)	0.0940 (0.0241)	27.12
10	0.0149 (0.0008)	0.0873 (0.0247)	0.0901 (0.0241)	56.61
20	0.0100 (0.0005)	0.0885 (0.0248)	0.0896 (0.0245)	111.03
50	0.0064 (0.0003)	0.0878 (0.0246)	0.0884 (0.0245)	281.39

5.3 Distribution of Split-Specific Estimates

To visualize split-induced computational variability directly, Figure 4 presents the empirical distribution of DML estimates obtained from repeated randomized cross-fitting executions on a fixed dataset. Although the underlying dataset, nuisance learners and model specification remain unchanged across repetitions, the resulting DML estimates exhibit noticeable dispersion due solely to randomized fold assignment. The histogram demonstrates that split-induced fluctuations remain practically observable even in moderate sample sizes. The resulting estimator distribution spans a nontrivial range, despite the absence of additional sampling variability. This visualization therefore provides direct computational evidence that randomized cross-fitting introduces an additional layer of algorithmic uncertainty beyond classical statistical sampling variability.

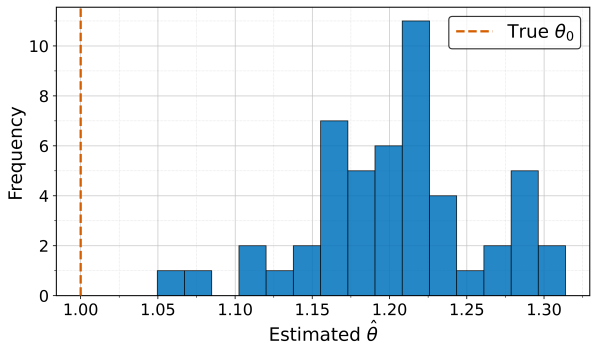
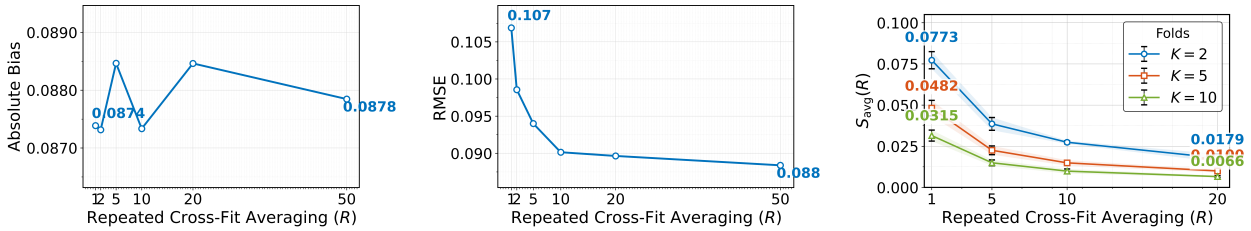


Figure 4: Distribution of split-specific DML estimates under repeated randomized fold assignments on a fixed dataset

5.4 Bias and RMSE Behavior

To better understand the effect of repeated averaging on estimator performance, Figures 5a and 5b report the empirical bias and RMSE across repeated cross-fitting repetitions. The experiments indicate that repeated cross-fitting has relatively little effect on estimator bias. Across all repetition counts, the empirical bias remains approximately constant. In contrast, RMSE decreases substantially as the number of repetitions increases. Since RMSE combines both variance and bias contributions, the observed decrease is primarily attributable to reduction of split-induced variance. These findings provide additional evidence that repeated cross-fitting behaves primarily as a computational variance reduction mechanism rather than a bias correction procedure.



(a) Absolute bias under repeated cross-fitting

(b) Root-mean-square error under repeated cross-fitting

(c) Fold configuration under repeated cross-fitting

Figure 5: Bias, RMSE behavior and fold configuration under repeated cross-fitting

5.5 High-Dimensional Instability

We next investigate split instability in a highly correlated high-dimensional setting with $(n, p, \rho) = (100, 500, 0.95)$. Table 3 and Figure 6 compare the baseline moderate-dimensional configuration with the highly correlated high-dimensional regime. The results indicate substantially increased instability in the high-dimensional setting. The instability increases from approximately 0.048 in the baseline regime to approximately 0.068 in the high-dimensional setting. This amplification is consistent with the increased difficulty of nuisance estimation in highly correlated finite-sample regimes. The experiments therefore suggest that split-induced computational variability becomes more pronounced precisely in settings where nuisance estimation is most difficult.

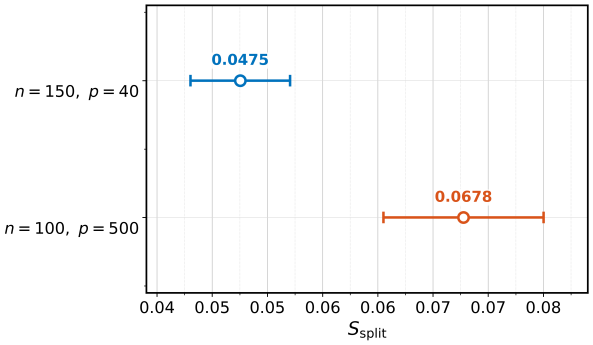


Figure 6: Split instability increases substantially in highly correlated high-dimensional settings.

5.6 Learner Comparisons

We next investigate the interaction between split instability, finite-sample bias and nuisance learner choice under the nonlinear nuisance specification described in Section 4. Table 4 and Figure 7 compare ridge,

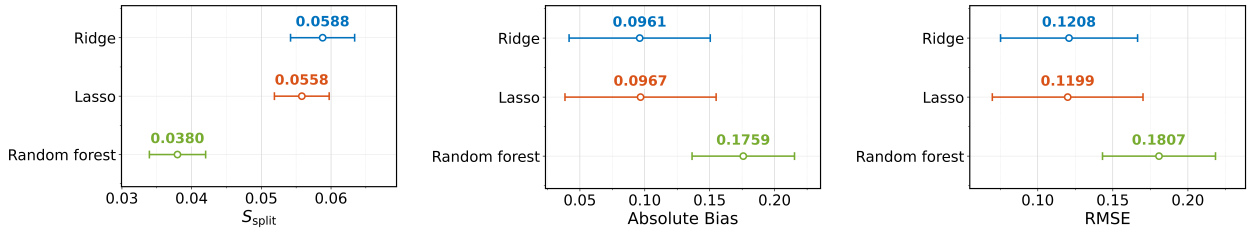
Table 3: High-dimensional instability comparison

n	p	ρ	Mean Instability	Monte Carlo Std. Error
150	40	0.8	0.0475	0.0023
100	500	0.95	0.0678	0.0037

Table 4: Learner comparison under nonlinear nuisance specification. Reported values are means across simulated datasets with standard errors in parentheses.

Learner	Instability	Absolute Bias	RMSE
Ridge	0.0588 (0.0023)	0.0961 (0.0277)	0.1208 (0.0233)
Lasso	0.0558 (0.0020)	0.0967 (0.0297)	0.1199 (0.0256)
Random Forest	0.0380 (0.0021)	0.1759 (0.0201)	0.1807 (0.0192)

lasso and random forest nuisance learners. The experiments reveal substantial variation across nuisance learners. Random forest nuisance estimators produce lower split instability relative to ridge and lasso learners but substantially larger finite-sample bias and RMSE. Ridge and lasso learners exhibit similar instability levels, although ridge regression produces slightly smaller finite-sample bias. In contrast, the random forest learner is associated with substantially larger estimator bias despite improved split stability. These findings suggest that lower split instability alone does not necessarily imply superior finite-sample performance. Instead, nuisance learner selection interacts nontrivially with orthogonalization quality, regularization bias and estimator stability.



(a) Split instability across nuisance learners

(b) Absolute bias across nuisance learners

(c) Root-mean-square error across nuisance learners

Figure 7: Learner comparison diagnostics

5.7 Sensitivity to the Number of Cross-Fitting Folds

We additionally investigate the sensitivity of split-induced instability to the number of cross-fitting folds. Table 5 and Figure 2b present the instability behavior for $K \in \{2, 5, 10, 20\}$. The reported experiments indicate that estimator instability decreases systematically as the number of folds increases. Specifically, the instability decreases from approximately 0.086 for $K = 2$ to approximately 0.020 for $K = 20$. This behavior likely reflects improved nuisance estimation resulting from the larger effective training samples available under larger fold counts. Although larger values of K improve stability in the present experiments, the magnitude of the gains decreases gradually for larger fold configurations. Overall, the experiments suggest that moderate-to-large fold counts provide favorable computational stability behavior in finite samples.

5.8 Interactions Between Fold Configuration and Repeated Cross-Fitting

We next investigate the interaction between fold configuration and repeated cross-fitting averaging. Table 6 and Figure 5c present the averaged-estimator instability measure across varying numbers of folds and repeated cross-fitting repetitions. Across all fold configurations, repeated cross-fitting consistently re-

Table 5: Sensitivity of split instability to the number of cross-fitting folds

Number of Folds (K)	Mean Instability	Monte Carlo Std. Error
2	0.0862	0.0033
5	0.0475	0.0023
10	0.0300	0.0013
20	0.0201	0.0008

Table 6: Interaction between fold configuration and repeated cross-fitting stabilization

Number of Folds (K)	Repetitions (R)	Mean Instability	Monte Carlo Std. Error
2	1	0.0773	0.0027
2	5	0.0386	0.0020
2	10	0.0274	0.0005
2	20	0.0179	0.0012
5	1	0.0482	0.0024
5	5	0.0226	0.0013
5	10	0.0149	0.0008
5	20	0.0100	0.0005
10	1	0.0315	0.0017
10	5	0.0149	0.0009
10	10	0.0099	0.0007
10	20	0.0066	0.0003

duces estimator instability. The overall stabilization behavior remains approximately consistent with inverse square-root variance reduction. However, the magnitude of the instability differs across fold configurations, indicating that fold selection and repeated averaging jointly influence computational stability. The experiments suggest that repeated cross-fitting stabilization remains robust across a broad range of practically relevant fold configurations.

5.9 No-Crossfit Comparisons

Finally, we compare no-crossfit estimation, single cross-fitting and repeated cross-fitting procedures. Table 7 and Figure 8 present the corresponding comparisons. The experiments reveal a nuanced tradeoff between orthogonalized cross-fitting and finite-sample computational variability. In the reported setting, the no-crossfit estimator achieves slightly lower empirical RMSE than the cross-fitted procedures, while single cross-fitting exhibits the largest RMSE due to additional split-induced variability. Repeated cross-fitting reduces RMSE relative to single cross-fitting through averaging across independently randomized fold assignments.

These findings suggest that repeated cross-fitting successfully stabilizes cross-fitted estimation procedures by reducing computational variance. At the same time, the results illustrate that orthogonalized cross-fitting may introduce additional finite-sample variability relative to procedures that avoid sample splitting entirely. Consequently, the benefits of cross-fitting should be interpreted jointly in terms of statistical robustness, orthogonality properties and computational stability. Overall, the experiments consistently demonstrate that randomized sample splitting induces measurable finite-sample computational variability in cross-fitted DML estimators, while repeated cross-fitting substantially improves estimator stability and reproducibility through computational variance reduction effects.

6 Discussion

This paper investigated the computational stability of cross-fitted DML estimators under repeated randomized sample splitting. While existing DML research has primarily emphasized asymptotic efficiency, orthogonal score construction and statistical consistency, the present work focused on the finite-sample computational variability induced by randomized cross-fitting itself. The experiments demonstrate that randomized fold assignment introduces measurable estimator instability, even when the underlying dataset, nuisance learners and model specification remain fixed. Repeated cross-fitting substantially reduces this instability

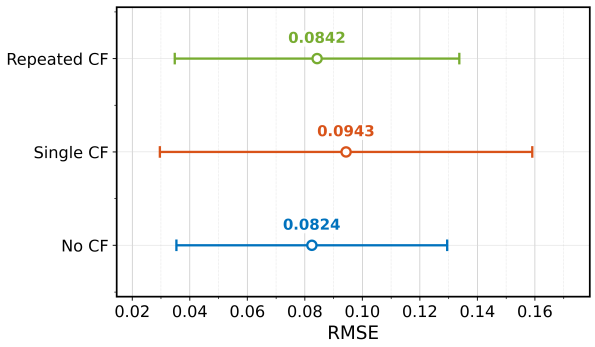


Figure 8: Root-mean-square error comparison across no-crossfit, single-crossfit and repeated-crossfit procedures

Table 7: Comparison of no-crossfit, single-crossfit and repeated-crossfit procedures

Method	Mean Estimate	Absolute Bias	RMSE	Std. Across Datasets
No CF	1.0540	0.0540	0.0824	0.1004
Single CF	1.0539	0.0539	0.0943	0.1326
Repeated CF	1.0570	0.0570	0.0842	0.1032

and empirically exhibits behavior approximately consistent with inverse square-root variance reduction under repeated averaging.

From a practical perspective, these findings suggest that single-split DML estimates may be sensitive to particular realizations of randomized fold assignments, especially in moderate sample sizes and difficult nuisance estimation regimes. Repeated cross-fitting therefore provides a simple stabilization strategy that improves robustness and reproducibility without modifying the underlying estimation procedure. The experiments further indicate that this stabilization behavior remains qualitatively consistent across varying sample sizes, fold configurations, high-dimensional settings and nuisance learners.

One important observation is that split-induced instability represents a computational component of uncertainty that is distinct from classical statistical sampling variability. Standard asymptotic confidence intervals quantify uncertainty arising from random sampling, whereas the present experiments isolate variability induced solely by randomized algorithmic implementation choices. In moderate sample sizes, this additional source of variability may therefore remain practically relevant for reproducibility considerations.

The experiments additionally reveal that lower split instability does not necessarily imply superior finite-sample performance. In particular, random forest nuisance estimators produced lower instability but substantially larger finite-sample bias and RMSE relative to ridge and lasso learners. Similarly, although repeated cross-fitting improved RMSE relative to single cross-fitting through variance reduction, the no-crossfit baseline occasionally achieved lower empirical RMSE in moderate sample sizes. These findings suggest that computational stability should be interpreted jointly with classical estimation metrics such as bias, RMSE and inferential robustness.

The runtime experiments indicate that repeated cross-fitting introduces approximately linear computational overhead in the number of repetitions. Nevertheless, the observed runtime growth remained moderate in the reported experiments, suggesting a practical tradeoff between estimator stability and computational efficiency.

Overall, the results demonstrate that randomized sample splitting induces measurable finite-sample computational variability in cross-fitted DML estimators and that repeated cross-fitting provides a practical stabilization strategy. More broadly, the study highlights the importance of computational reproducibility diagnostics in modern ML-assisted inference procedures that rely on randomized algorithmic components.

7 Limitations and Future Work

Several limitations of the present study should be acknowledged.

First, the experiments primarily focused on synthetic partially linear regression settings, in order to isolate the computational effects of randomized cross-fitting. Although controlled simulations are useful for studying estimator stability under reproducible conditions, the quantitative behavior of split-induced variability may differ in observational datasets and practical causal inference applications.

Second, the experiments considered a relatively limited collection of nuisance learners, namely ridge regression, lasso regression and random forests. More complex learners, including boosting methods, deep neural networks and transformer-based architectures, may exhibit different stability properties and stronger sensitivity to algorithmic randomness.

Third, the present work does not provide formal theoretical analysis for the proposed instability metrics. The observed inverse square-root stabilization behavior under repeated cross-fitting was established empirically through simulation rather than derived theoretically.

Another limitation concerns experimental scale. The experiments were intentionally designed to remain lightweight and reproducible on standard hardware. Larger simulation studies may provide additional insight into the interaction between sample size, nuisance complexity and algorithmic instability.

The present work additionally focused on standard partially linear regression models. Extensions to more complex semiparametric and causal inference settings, including heterogeneous treatment effect estimation, instrumental variable models and policy learning frameworks, remain important directions for future research.

Several methodological extensions also emerge naturally from the present study. These include theoretical analysis of split-induced computational variability, adaptive procedures for selecting the number of repeated cross-fitting repetitions and broader investigation of how algorithmic randomness interacts with statistical uncertainty in ML-assisted inference pipelines.

Finally, additional empirical validation on observational datasets represents an important next step. Real-data applications may reveal reproducibility issues not fully captured by controlled simulation studies and may further clarify the practical importance of computational stability diagnostics in applied inference workflows.

8 Conclusion

This paper studied the finite-sample computational stability of cross-fitted DML estimators under repeated randomized sample splitting. The experiments demonstrated that randomized fold assignment induces measurable estimator variability, even when the underlying dataset and model specification remain fixed.

Across a broad collection of synthetic experiments, repeated cross-fitting substantially improved estimator stability and exhibited empirical behavior approximately consistent with inverse-square-root stabilization behavior under repeated averaging. The experiments further showed that split-induced instability decreases with increasing sample size, becomes amplified in highly correlated high-dimensional regimes, varies with the number of cross-fitting folds and interacts nontrivially with nuisance learner choice.

The empirical results additionally suggest that repeated cross-fitting primarily improves estimator performance through reduction of split-induced variance rather than estimator bias. While repeated averaging consistently reduced instability and improved RMSE relative to single cross-fitting, the experiments also highlighted finite-sample tradeoffs between orthogonalized cross-fitting and procedures that avoid sample splitting entirely.

Overall, the results suggest that computational randomness introduced by cross-fitting represents a practically relevant component of uncertainty in finite-sample ML-assisted inference procedures. Consequently, repeated cross-fitting and instability diagnostics may provide useful tools for improving robustness, transparency and reproducibility in applied DML workflows.

More broadly, the study highlights the importance of computational stability analysis in modern ML-assisted inference procedures that rely heavily on randomized algorithmic components. Understanding how algorithmic randomness interacts with statistical uncertainty therefore remains an important direction for future research.

Broader Impact Statement

This work studies computational stability and reproducibility in ML-assisted statistical inference procedures. The proposed instability diagnostics and repeated cross-fitting methodology are intended to improve robustness and transparency in practical DML workflows. The paper does not introduce new data collection procedures, surveillance systems or high-risk deployment applications. The proposed methods are methodological and computational in nature and are primarily intended for statistical estimation and causal inference settings.

One potential risk is that computational stability diagnostics could be misinterpreted as substitutes for standard statistical uncertainty quantification. The proposed methods are intended to complement rather than replace classical inference procedures such as confidence intervals, asymptotic variance estimation and sensitivity analysis.

More broadly, the study highlights that randomized computational procedures may introduce practically measurable uncertainty in finite-sample ML-assisted inference pipelines. Making such variability explicit may encourage more reproducible and transparent reporting practices in randomized estimation procedures.

References

Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

Philipp Bach, Victor Chernozhukov, and Malte Kurz. DoubleML – An object-oriented implementation of double machine learning in Python. *Journal of Machine Learning Research*, 23(53):1–6, 2022.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, 2011.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

Victor Chernozhukov, Philipp Bach, Malte Kurz, et al. DoubleML: Double machine learning in Python. *Journal of Open Source Software*, 7(75):3783, 2022.

Bradley Efron and Robert Tibshirani. *An Introduction to the Bootstrap*. CRC Press, 1994.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2 edition, 2009.

John D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

Wes McKinney. Data structures for statistical computing in Python. pp. 56–61, 2010.

Whitney Newey and James Robins. Cross-fitting and fast remainder rates for semiparametric estimation. *arXiv preprint arXiv:1801.09138*, 2018.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Vladimir Vapnik. *Statistical Learning Theory*. Wiley, 1998.

Huan Xu and Shie Mannor. Robustness and generalization. *Machine Learning*, 86(3):391–423, 2012.

A Additional Experimental Details

Unless otherwise stated, all experiments used the following baseline configuration:

- Sample size: $n = 150$
- Covariate dimension: $p = 40$
- Correlation parameter: $\rho = 0.8$
- Number of folds: $K = 5$
- True treatment effect: $\theta_0 = 1$
- Monte Carlo datasets: 10
- Randomized split repetitions per dataset: 20

Repeated cross-fit estimators were evaluated for

$$R \in \{1, 2, 5, 10, 20, 50\} ,$$

while fold-sensitivity experiments used

$$K \in \{2, 5, 10, 20\} .$$

B Implementation and Reproducibility

All experiments were implemented in Python using NumPy (Harris et al., 2020), pandas (McKinney, 2010), scikit-learn (Pedregosa et al., 2011), matplotlib (Hunter, 2007) and DoubleML (Bach et al., 2022; Chernozhukov et al., 2022).

Randomized fold assignments were externally controlled using fixed random seeds. Random forest nuisance learners additionally used fixed internal seeds to isolate instability arising specifically from randomized cross-fitting.

All figures and tables were generated automatically from deterministic simulation pipelines whenever possible.

C Code Availability

The complete implementation, including simulation scripts, plotting utilities, repeated cross-fitting procedures and reproducibility workflows, will be made publicly available upon publication.