Manuscript submitted to *The Econometrics Journal*, pp. ??–??.

Finite-Sample Failures and Condition-Number Diagnostics in Double Machine Learning: Online Supplement

Gabriel Saco[†]

[†] *(Affiliation to be added)*
E-mail: gsacoalvarado@gmail.com

Summary

This online supplement contains additional theoretical details, extended simulation results, and supplementary proofs for "Finite-Sample Failures and Condition-Number Diagnostics in Double Machine Learning."

Keywords: *Double Machine Learning, Online Supplement.*

S1. ADDITIONAL THEORETICAL DETAILS

This section provides additional technical details supporting the main theoretical results.

S1.1. Detailed Proof of Lemma 3.2 (Refined Linearisation)

Proof: The DML estimator $\hat{\theta}$ solves $\Psi_n(\hat{\theta}, \hat{\eta}) = 0$. Using the PLR score (3.5), this becomes

$$\frac{1}{n}\sum_{i=1}^{n} \hat{U}_i(\hat{V}_i - \hat{\theta}\hat{U}_i) = 0, \tag{S.1}$$

which yields the closed-form solution (3.9). To derive the linearisation, write

$$\hat{\theta} - \theta_0 = \frac{\sum_i \hat{U}_i\hat{V}_i}{\sum_i \hat{U}_i^2} - \theta_0 = \frac{\sum_i \hat{U}_i(\hat{V}_i - \theta_0\hat{U}_i)}{\sum_i \hat{U}_i^2}. \tag{S.2}$$

Define population residuals $U_i := D_i - m_0(X_i)$ and $V_i := Y_i - \ell_0(X_i)$. By the model (3.2), $V_i = \theta_0 U_i + \varepsilon_i$ where $\mathbb{E}[\varepsilon_i \mid X_i, D_i] = 0$. The numerator decomposes as

$$\sum_i \hat{U}_i(\hat{V}_i - \theta_0\hat{U}_i) = \underbrace{\sum_i U_i\varepsilon_i}_{=:nS_n} + \underbrace{\sum_i \hat{U}_i(\hat{V}_i - V_i) - \theta_0\sum_i \hat{U}_i(\hat{U}_i - U_i)}_{=:nB_n} + R_n', \tag{S.3}$$

where R_n' collects higher-order cross-terms.

Analysis of S_n: By construction, $S_n = n^{-1}\sum_i U_i\varepsilon_i$ is a sample average of mean-zero random variables with variance σ_ψ^2/n. Under Assumption 3.1(iii), the CLT gives $\sqrt{n}S_n \xrightarrow{d} N(0, \sigma_\psi^2)$, so $S_n = O_P(n^{-1/2})$.

Analysis of B_n: The bias term B_n arises from nuisance estimation error. Expanding using $\hat{V}_i - V_i = (\ell_0(X_i) - \hat{\ell}(X_i))$ and $\hat{U}_i - U_i = (\hat{m}(X_i) - m_0(X_i))$, and applying orthogonality (Assumption 3.1(ii)) together with the product-rate condition (Assumption 3.3(i)), we obtain $B_n = O_P(r_n) = o_P(n^{-1/2})$.

Denominator: By Assumption 3.3(ii)–(iii), $n^{-1}\sum_i \hat{U}_i^2 \xrightarrow{p} \sigma_U^2$, so

$$\kappa_{\mathrm{DML}} = \frac{n}{\sum_i \hat{U}_i^2} = \frac{1}{n^{-1}\sum_i \hat{U}_i^2} \xrightarrow{p} \sigma_U^{-2}. \tag{S.4}$$

Combining these elements:

$$\hat{\theta} - \theta_0 = \frac{n(S_n + B_n) + R_n'}{\sum_i \hat{U}_i^2} = \kappa_{\mathrm{DML}}(S_n + B_n) + R_n, \tag{S.5}$$

where $R_n = o_P(n^{-1/2})$ absorbs remainder terms. The rate $R_n = o_P(n^{-1/2})$ follows from the product-rate condition ensuring $\kappa_{\mathrm{DML}} \cdot r_n = o_P(n^{-1/2})$ when $\kappa_{\mathrm{DML}} = O_P(1)$. □

S1.2. Proof of Proposition 3.3 (Efficiency Bound Connection)

Proof: The efficiency bound follows from Hahn (1998) and Hirano, Imbens, and Ridder (2003) applied to the PLR model. In this model, the influence function is $\psi(W;\theta_0,\eta_0) = U\varepsilon$, which has variance $\mathbb{E}[U^2\varepsilon^2]$. The Jacobian is $J_\theta = -\mathbb{E}[U^2]$, so

$$V_{\mathrm{eff}} = \frac{\mathbb{E}[\psi^2]}{J_\theta^2} = \frac{\mathbb{E}[U^2\varepsilon^2]}{(\mathbb{E}[U^2])^2}. \tag{S.6}$$

From Lemma 3.2, $\sqrt{n}(\hat{\theta} - \theta_0) = \sqrt{n}\kappa_{\mathrm{DML}} S_n + o_P(1)$, where $\sqrt{n}S_n \xrightarrow{d} N(0, \mathbb{E}[U^2\varepsilon^2])$. Since $\kappa_{\mathrm{DML}} \xrightarrow{p} 1/\mathbb{E}[U^2]$, the asymptotic variance is $\mathbb{E}[U^2\varepsilon^2]/(\mathbb{E}[U^2])^2 = V_{\mathrm{eff}}$, confirming that DML achieves the efficiency bound under good conditioning. The standard error expression follows from the plug-in estimator for V_{eff}. □

S2. EXTENDED SIMULATION RESULTS

This section provides additional Monte Carlo results beyond those reported in the main text.

S2.1. Detailed Design Description

We work in the PLR model with $n \in \{500, 2000\}$ observations, $p = 10$ covariates, and the following data-generating process:

(a) Covariates: $X \sim N(0, \Sigma)$ where $\Sigma_{jk} = \rho^{|j-k|}$ with $\rho = 0.5$.

(b) Treatment: $D = X'\beta_D + \sigma_U \cdot \nu$ where $\nu \sim N(0,1)$ is independent of X, $\beta_D = (1, 0.5, 0.25, \ldots)$, and σ_U^2 is calibrated to achieve target $R^2(D \mid X) \in \{0.75, 0.90, 0.97\}$.

(c) Outcome: $Y = D\theta_0 + g_0(X) + \varepsilon$ where $\theta_0 = 1$, $g_0(X) = X'\beta_Y$ with $\beta_Y = (0.5, 0.3, 0.2, \ldots)$, and $\varepsilon \sim N(0,1)$.

S2.2. Full Results Tables

Table S.1 provides the complete simulation results for all combinations of overlap level, sample size, and nuisance learner.

Table S.1. Complete Monte Carlo Results: PLR DML Simulations

R^2	Learner	n	$\bar{\kappa}$	Coverage	CI Len	Bias	RMSE	$\bar{R}^2$
0.75	LIN	500	0.67	95.2	0.18	0.001	0.046	0.75
	LIN	2000	0.66	95.0	0.09	0.000	0.023	0.75
	LAS	500	0.67	94.6	0.18	0.001	0.047	0.75
	LAS	2000	0.66	94.8	0.09	0.000	0.024	0.75
	RF	500	0.68	88.2	0.16	−0.025	0.048	0.75
	RF	2000	0.66	89.8	0.08	−0.012	0.025	0.75
0.90	LIN	500	1.70	95.4	0.31	0.001	0.080	0.90
	LIN	2000	1.66	95.0	0.16	0.001	0.040	0.90
	LAS	500	1.71	94.4	0.31	0.002	0.082	0.90
	LAS	2000	1.66	95.2	0.16	0.000	0.041	0.90
	RF	500	1.73	76.4	0.25	−0.058	0.086	0.90
	RF	2000	1.67	79.6	0.13	−0.032	0.046	0.90
0.97	LIN	500	5.12	94.6	0.59	−0.003	0.152	0.97
	LIN	2000	4.53	95.0	0.30	−0.001	0.077	0.97
	LAS	500	5.15	93.8	0.59	−0.002	0.156	0.97
	LAS	2000	4.54	94.4	0.30	−0.001	0.079	0.97
	RF	500	5.28	66.2	0.35	−0.098	0.134	0.97
	RF	2000	4.58	70.0	0.18	−0.062	0.078	0.97

Notes: $B = 500$ replications per cell. LIN = linear regression, LAS = Lasso, RF = random forest. $\bar{\kappa}$ = mean κ_{DML}, $\bar{R}^2$ = mean sample $R^2(D \mid X)$. Coverage is the proportion of nominal 95% CIs containing $\theta_0 = 1$. CI Len = average confidence interval length.

S3. DATA SOURCES AND REPLICATION

S3.1. LaLonde (1986) Data

The empirical application uses data from the National Supported Work (NSW) demonstration, as analysed by LaLonde (1986). We use two samples:

(a) **Experimental sample** ($n = 445$): The original randomised experiment combining NSW treated units ($n = 185$) with NSW control units ($n = 260$).
(b) **Observational sample** ($n = 2,675$): NSW treated units ($n = 185$) combined with the PSID-1 comparison group ($n = 2,490$).

The outcome variable is real earnings in 1978 (in 1982 dollars). Covariates include age, years of education, indicators for Black and Hispanic ethnicity, marital status, high school diploma, and lagged earnings in 1974 and 1975.

S3.2. Replication Code

All simulations and empirical analyses were conducted in Python using the `dml_diagnostic` package. Replication code is available at `https://github.com/gsaco/dml-diagnostic`.

S4. THE `DML_DIAGNOSTIC` PYTHON PACKAGE

This section documents the `dml_diagnostic` Python package, which implements the DML condition number diagnostic κ_{DML} for practitioners.

S4.1. Installation

The package can be installed via pip:

```
pip install dml-diagnostic
```

For the latest development version:

```
pip install git+https://github.com/gsaco/dml-diagnostic.git
```

Dependencies: numpy $\geq$ 1.20, pandas $\geq$ 1.3, scikit-learn $\geq$ 1.0. Optional: matplotlib for plotting.

S4.2. Quick Start

The package provides a simple API for DML estimation with condition number diagnostics:

```
from dml_diagnostic import DMLDiagnostic, load_lalonde

# Load LaLonde experimental data
Y, D, X = load_lalonde(sample='experimental')

# Fit DML estimator with diagnostics
dml = DMLDiagnostic(learner='lasso')
result = dml.fit(Y, D, X)

# Print results with interpretation
print(result)
```

Output:

```
DML Diagnostic Results
----------------------
  theta = 1793.42 (SE = 672.45)
  95% CI: [475.41, 3111.43]

  Condition Number: kappa_DML = 4.10

  n = 445, R^2(D|X) = -0.003, learner = lasso
```

S4.3. API Reference

DMLDiagnostic class. Main estimator with condition number diagnostics.

```
DMLDiagnostic(
    learner='lasso',    # 'lin', 'lasso', 'ridge', 'rf', 'gbm'
    learner_m=None,     # Separate learner for E[D|X]
    learner_g=None,      # Separate learner for E[Y|X]
    n_folds=5,           # Cross-fitting folds
    random_state=42      # For reproducibility
)
```

Methods:

- `fit(Y, D, X)`: Fit DML and compute κ_{DML}, returns `DMLResult`.
- `summary()`: Print detailed results with interpretation.

DMLResult attributes.

- `theta`: Point estimate $\hat{\theta}$.
- `se`: Standard error.
- `ci_lower`, `ci_upper`: 95% CI bounds.
- `kappa`: Condition number κ_{DML}.
- `jacobian`: Empirical Jacobian $\hat{J}_\theta$.
- `r_squared_d`: $R^2(D \mid X)$ from treatment regression.
- `U_hat`, `V_hat`: Cross-fitted residuals.

Data loading.

```
load_lalonde(
    sample='experimental',  # or 'observational'
    return_dataframe=False, # True returns DataFrame
    verbose=False
)
```

Returns `(Y, D, X)` arrays for the LaLonde (1986) dataset.

Diagnostic functions.

```
# Compute kappa from treatment residuals
compute_kappa(U_hat, n=None)

# Contextual interpretation
kappa_interpretation(kappa, n, r_squared_d=None)

# Overlap diagnostics via propensity scores
overlap_check(D, X, method='logistic')
```

S4.4. Comparing Experimental and Observational Samples

The following example illustrates how κ_{DML} captures the difference in conditioning between the experimental and observational LaLonde samples:

```
from dml_diagnostic import DMLDiagnostic, load_lalonde
```

```
# Experimental sample (good overlap)
Y_exp, D_exp, X_exp = load_lalonde('experimental')
result_exp = DMLDiagnostic(learner='lasso').fit(Y_exp, D_exp, X_exp)

# Observational sample (poor overlap)
Y_obs, D_obs, X_obs = load_lalonde('observational')
result_obs = DMLDiagnostic(learner='lasso').fit(Y_obs, D_obs, X_obs)

print(f"Experimental: theta = {result_exp.theta:.0f}, kappa = {result_exp.kappa:.2f}")
print(f"Observational: theta = {result_obs.theta:.0f}, kappa = {result_obs.kappa:.2f}")
```

Output:

```
Experimental: theta = 1793, kappa = 4.10
Observational: theta = 56, kappa = 15.71
```

The higher κ_{DML} in the observational sample reflects the poor overlap between NSW treated units and PSID controls: treatment is more predictable from covariates, leaving less residual variation for identification. The experimental benchmark estimate of approximately \$1,800 is recovered when conditioning is good, but the observational estimate is unreliable due to the flat score.

REFERENCES

Bach, P., V. Chernozhukov, M. S. Kurz and M. Spindler (2022). DoubleML – An object-oriented implementation of double machine learning in Python. *Journal of Machine Learning Research 23*(53), 1–6.

Hahn, J. (1998). On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica 66*(2), 315–331.

Hirano, K., G. W. Imbens and G. Ridder (2003). Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica 71*(4), 1161–1189.

LaLonde, R. J. (1986). Evaluating the econometric evaluations of training programs with experimental data. *American Economic Review 76*(4), 604–620.