# Debiasing Randomized Numerical Linear Algebra via Random Matrix Theory

International Conference on Applied Mathematics 2026 (ICAM 2026)

**Zhenyu Liao**

Huazhong University of Science & Technology (HUST)

Joint work with Chengmei Niu (HUST), Sachin Garg (Umich), Michał Dereziński (Umich), Edgar Dobriban (UPenn), and Michael W. Mahoney (UC Berkeley)

June 10, 2026

# One-slide summary

For many (R)NLA problems, unbiasedness alone is **not** enough.

For a random sketch $\widetilde{\mathbf{X}} = \mathbf{S}\mathbf{X}$ with $\mathbb{E}[\mathbf{S}^\top \mathbf{S}] = \mathbf{I}_n$, one have

$$\mathbb{E}[\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}}] = \mathbf{X}^\top \mathbf{X}, \quad \text{and} \quad \widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}} \approx \mathbf{X}^\top \mathbf{X},$$

but the nonlinear quantities used by RNLA algorithms are generally **biased**:

$$\mathbb{E}[(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1}] \neq (\mathbf{X}^\top \mathbf{X})^{-1}, \qquad \mathbb{E}\big[\mathbf{X}(\mathbf{S}\mathbf{X})^\dagger \mathbf{S}\big] \neq \mathbf{X}\mathbf{X}^\dagger.$$

**RMT viewpoint**

RMT provides a way to characterize and debias these nonlinear random operators.

# Numerical linear algebra at scale

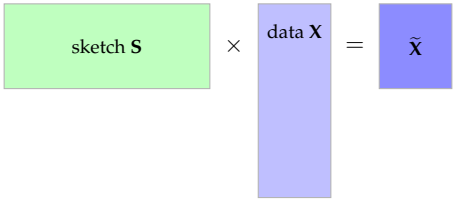
From numerical linear algebra (NLA) to RandNLA:

- of daily use in scientific computing and **machine learning** (ML)
- includes least squares, low-rank approximation, SVD, optimization, etc.
- in many ML applications, high (e.g., machine) precision is not needed
- the central goal is a better trade-off between precision and computational complexity
- randomness can play a central role: RandNLA

# Sketch-and-solve

For a tall matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ with $n \gg d$, construct a sketch

$$\widetilde{\mathbf{X}} = \mathbf{S}\mathbf{X} \in \mathbb{R}^{m\times d}, \qquad m \ll n.$$




- **Random projection**: Gaussian, sub-gaussian, SRHT/SRFT.
- **Random sampling**: uniform, row-norm, leverage score, approximate leverage score.

# Johnson–Lindenstrauss lemma

Classical RandNLA perspective starts from the Johnson–Lindenstrauss principle.

## Johnson–Lindenstrauss (JL) lemma, [JL84]

For any $d$ points $\mathbf{a}_1, \ldots, \mathbf{a}_d \in \mathbb{R}^n$ and any $\varepsilon \in (0, 1/2)$, there exists a random linear map $\mathbf{S} : \mathbb{R}^n \to \mathbb{R}^m$ with $\mathbb{E}[\mathbf{S}^\top \mathbf{S}] = \mathbf{I}_n$ and

$$m \gtrsim \log d / \varepsilon^2$$

such that, with high probability, for all $i, j$,

$$(1-\varepsilon)\|\mathbf{a}_i - \mathbf{a}_j\|^2 \leq \|\mathbf{S}(\mathbf{a}_i - \mathbf{a}_j)\|^2 \leq (1+\varepsilon)\|\mathbf{a}_i - \mathbf{a}_j\|^2$$

## From finite sets to subspaces

For $\tilde{\mathbf{X}} = \mathbf{S}\mathbf{X}$, applying JL to a finite net of $\mathrm{col}(\mathbf{X})$ gives a subspace embedding for $\mathbf{X}$.

[1] William B. Johnson and Joram Lindenstrauss. "Extensions of Lipschitz mappings into a Hilbert space". In: *Contemporary Mathematics* (1984), pp. 189–206

# The standard guarantee: subspace embedding

## Subspace embedding

A sketch $\widetilde{\mathbf{X}} = \mathbf{S}\mathbf{X}$ is an $(\varepsilon, \delta)$-subspace embedding for $\mathbf{X}$ if, with probability at least $1 - \delta$,

$$(1+\varepsilon)^{-1}\mathbf{X}^\top\mathbf{X} \preceq \widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}} \preceq (1+\varepsilon)\mathbf{X}^\top\mathbf{X}.$$

- controls all quadratic forms $\|\mathbf{S}\mathbf{X}\boldsymbol{\beta}\|^2$ (uniformly in $\boldsymbol{\beta}$)
- explains why sketch-and-solve approaches work (at least to some extend)
- in essence: first-order moment control + high-probability concentration result

# Why subspace embedding is NOT enough

Many downstream algorithms use nonlinear functions of the sketch:

$$\text{inverse Gram matrix:} \quad (\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1} = (\mathbf{X}^\top \mathbf{S}^\top \mathbf{S}\mathbf{X})^{-1},$$
$$\text{oblique projection:} \quad \widetilde{\mathbf{P}} = \mathbf{X}(\mathbf{S}\mathbf{X})^\dagger \mathbf{S}.$$

## The missing question

- JL/subspace embedding ensures $\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}} \simeq \mathbf{X}^\top \mathbf{X}$ in some matrix norm sense
- but does not precisely characterize

$$\mathbb{E}[(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1}] \qquad \text{or} \qquad \mathbb{E}[\widetilde{\mathbf{P}}]$$

# Inversion bias

For $\widetilde{\mathbf{X}} = \mathbf{S}\mathbf{X}$, even when

$$\mathbb{E}[\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}}] = \mathbf{X}^\top \mathbf{X},$$

one generally has

$$\mathbb{E}[(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1}] \neq (\mathbf{X}^\top \mathbf{X})^{-1}.$$

- this is not surprising: inverse is nonlinear
- but algorithmically consequential: randomized least squares, Newton method, etc.
- this **bias** often invisible to coarse JL-style analyses:
  $\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}} \approx \mathbf{X}^\top \mathbf{X} \Rightarrow (\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1} \approx (\mathbf{X}^\top \mathbf{X})^{-1}$

# A familiar scalar case: Gaussian random projection

If $\mathbf{S} \in \mathbb{R}^{m\times n}$ has i.i.d. $\mathcal{N}(0, 1/m)$ entries, then the mean inverse-Wishart distribution gives

$$\mathbb{E}[(\mathbf{X}^\top \mathbf{S}^\top \mathbf{S}\mathbf{X})^{-1}] = \frac{m}{m-d-1}(\mathbf{X}^\top \mathbf{X})^{-1}.$$

- inversion bias is a scalar inflation
- a simple multiplicative correction is possible
- similar scalar correction $\frac{m}{m-d}$ remains effective for random projections, e.g., sub-gaussian and LESS [Der+21]

**RMT note**

Reminiscent of the **resolvent** of sample covariance matrix $(\mathbf{X}^\top \mathbf{S}^\top \mathbf{S}\mathbf{X} - z\mathbf{I}_d)^{-1}$ at $z = 0$.

But **random sampling** is different: the bias is generally matrix-valued.

[2] Michal Derezinski, **Z. Liao**, Edgar Dobriban, and Michael Mahoney. "Sparse sketches with small inversion bias". In: *Proceedings of Thirty Fourth Conference on Learning Theory*. Vol. 134. PMLR, 15–19 Aug 2021, pp. 1467–1510

# Random sampling: leverage controls the bias

For $\mathbf{X} \in \mathbb{R}^{n\times d}, n \gg d$, consider row sampling with probabilities $\{\pi_i\}_{i=1}^n$, **leverage scores**:

$$\ell_i(\mathbf{X}) = \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i, \qquad \theta_i = \frac{\ell_i(\mathbf{X})}{d\pi_i}.$$

- exact **leverage score sampling** [Mah11]: $\theta_i = 1$ for all $i$ (and $\theta_i \approx 1$ for approx leverage sampling)
- **note**: uniform sampling may have very **nonuniform** $\theta_i$ when leverage scores of $\mathbf{X}$ are highly nonuniform

**Inversion bias for random sampling, [Niu+25]**

Inverse bias for random sampling depends on the leverage scores profile, not only on $d/m$

# Matrix debiasing for random sampling

RMT analysis suggests that sampling should be debiased at the **matrix level**:

$$\check{\mathbf{S}} = \operatorname{diag}\left\{\sqrt{\frac{m}{m-\ell_{i_s}(\mathbf{X})/\pi_{i_s}}}\right\}_{s=1}^{m}\mathbf{S}.$$

## Matrix debiasing for random sampling, [Niu+25]

With appropriate sample size $m$ and conditioning on a high-probability event $\zeta$,

$$\mathbb{E}_\zeta\left[(\mathbf{X}^\top\check{\mathbf{S}}^\top\check{\mathbf{S}}\mathbf{X})^{-1}\right] \simeq (\mathbf{X}^\top\mathbf{X})^{-1}.$$

- exact leverage score sampling: reduces to the SAME scalar $\frac{m}{m-d}$
- for general sampling, scalar correction is insufficient

[3] Chengmei Niu, **Z. Liao**, Zenan Ling, and Michael W. Mahoney. "Fundamental Bias in Inverting Random Sampling Matrices with Application to Sub-sampled Newton". In: *Proceedings of the 42nd International Conference on Machine Learning*. PMLR, Oct. 2025, pp. 46649–46692

# Application to debias subsampled Newton

- consider optimization problem:

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}\in\mathcal{C}} F(\boldsymbol{\beta}) = \arg\min_{\boldsymbol{\beta}\in\mathcal{C}} f(\boldsymbol{\beta}) + \Phi(\boldsymbol{\beta}),$$

  for smooth $F\colon \mathbb{R}^d \to \mathbb{R}$ decomposed into $f$ and $\Phi$, and convex $\mathcal{C} \subseteq \mathbb{R}^d$

- Newton's method:

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \mu_t \mathbf{H}_t^{-1}(\boldsymbol{\beta}_t)\mathbf{g}_t$$

  with $\mu_t$ the step size, $\mathbf{g}_t \in \mathbb{R}^d$ the gradient, and $\mathbf{H}_t(\boldsymbol{\beta}_t) \in \mathbb{R}^{d\times d}$ the Hessian of $F$ at $\boldsymbol{\beta}_t$ such that

$$\mathbf{H}_t(\boldsymbol{\beta}_t) = \mathbf{A}(\boldsymbol{\beta}_t)^\top \mathbf{A}(\boldsymbol{\beta}_t) + \mathbf{C}(\boldsymbol{\beta}_t),$$

  with $\mathbf{A}(\boldsymbol{\beta}_t) \in \mathbb{R}^{n\times d}$ and some p.s.d. simple matrix $\mathbf{C}(\boldsymbol{\beta}_t) = \nabla^2\Phi(\boldsymbol{\beta}_t) \in \mathbb{R}^{d\times d}$, e.g., $\mathbf{C}(\boldsymbol{\beta}_t) = 2\lambda\mathbf{I}_d$ in the case of $L_2$ regularization $\Phi(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}\|^2$.

# Connection to randomized second-order optimization

Subsampled Newton (SSN) uses

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \mu_t \underbrace{\Big(\mathbf{A}(\boldsymbol{\beta}_t)^\top \mathbf{S}_t^\top \mathbf{S}_t \mathbf{A}(\boldsymbol{\beta}_t) + \mathbf{C}(\boldsymbol{\beta}_t)\Big)^{-1}}_{} \mathbf{g}_t.$$

- update direction depends directly on sampled inverse Hessian
- inversion bias can **slow** local convergence
- matrix debiasing improves Hessian approximation under random sampling

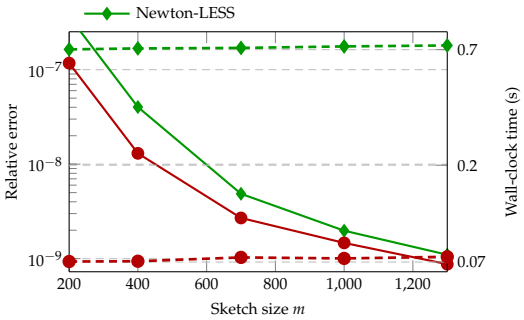
## Local convergence rate of debiased SSN, [Niu+25]

Debiased SSN recovers a local convergence rate of order

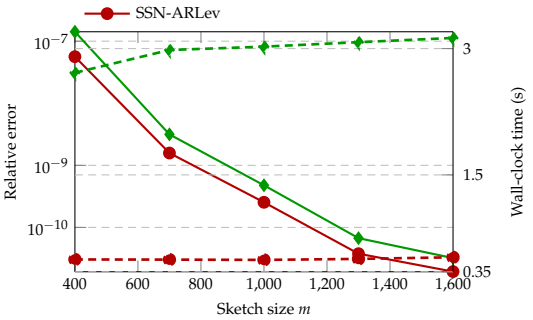
$$\frac{\theta_{\max} d_{\text{eff}}}{m}(1 + o(1)), \quad \theta_{\max} = \max_i \frac{\ell_i(\mathbf{X})}{d\pi_i},$$

and matches SoTA result (of Newton-LESS).

# Numerical results



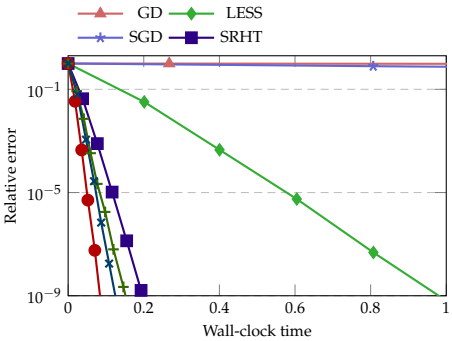

(a) MNIST data



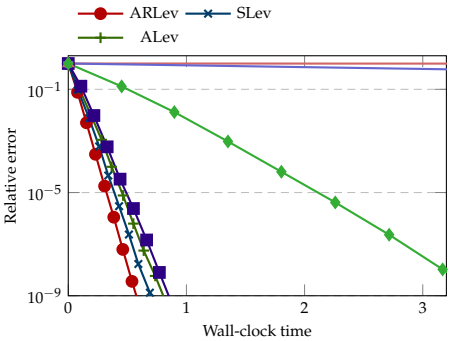

(b) CIFAR-10 data

Figure: Relative errors (in solid lines) and wall-clock time (in dashed lines) as a function of the sketch size $m$, for **Newton-LESS** and the proposed de-biased **SSN-ARLev** methods, applied to logistic regression on both MNIST and CIFAR-10 data. Relative errors are obtained after a fixed number of iterations ($T = 5$ for MNIST data and $T = 7$ for CIFAR-10 data). Results are obtained by averaging over 30 independent runs.

# Numerical results




(a) MNIST data




(b) CIFAR-10 data

Figure: Convergence–complexity trade-off between various optimization methods on logistic regression for MNIST and CIFAR-10 data, with sketch size $m = 300$ for MNIST and $m = 400$ for CIFAR-10 data. Results are obtained by averaging over 10 independent runs (except for GD that is deterministic).

# The random oblique projection

For $\mathbf{X} \in \mathbb{R}^{n\times d}, n \gg d$ of rank $d$, define

$$\mathbf{P} = \mathbf{X}\mathbf{X}^\dagger, \qquad \mathbf{P}_\perp = \mathbf{I}_n - \mathbf{P},$$

and the sampling-induced oblique projection

$$\widetilde{\mathbf{P}} = \mathbf{X}(\mathbf{S}\mathbf{X})^\dagger\mathbf{S},$$

for random sampling matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$.

## Why it matters

Subsampled OLS solution $\widetilde{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{S}\mathbf{y} - \mathbf{S}\mathbf{X}\boldsymbol{\beta}\|^2$ governed by $\widetilde{\mathbf{P}}$:

$$\mathbf{X}\widetilde{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{S}\mathbf{X})^\dagger\mathbf{S}\mathbf{y} = \widetilde{\mathbf{P}}\mathbf{y}.$$

# Debiased oblique projection

Use the same matrix-level debiased sampling matrix

$$\check{\mathbf{S}} = \operatorname{diag}\left\{\frac{1}{\sqrt{1-\ell_{i_s}(\mathbf{X})/(m\pi_{i_s})}}\right\}_{s=1}^{m}\mathbf{S}, \qquad \check{\mathbf{P}} = \mathbf{X}(\check{\mathbf{S}}\mathbf{X})^{\dagger}\check{\mathbf{S}}.$$

## First- and second-order characterizations of oblique projection, [Niu+26]

With appropriate sample size $m$ and conditioned on a high-probability event $\zeta$,

$$\|\mathbb{E}_{\zeta}[\check{\mathbf{P}}] - \mathbf{P}\|_F^2 = \varepsilon^2\|\mathbf{P}_{\perp}\|_F^2,$$

$$\mathbb{E}_{\zeta}\|\check{\mathbf{P}} - \mathbf{P}\|_F^2 = \operatorname{tr}\left(\mathbf{P}_{\perp}\operatorname{diag}\{\ell_i(\mathbf{X})/(m\pi_i)\}_{i=1}^{n}\right) + \varepsilon\|\mathbf{P}_{\perp}\|_F^2,$$

for $\varepsilon = \tilde{O}(\sqrt{\theta_{\max}^3 d^3/m^3})$.

[3] Chengmei Niu, Sachin Garg, Michał Dereziński, and **Z. Liao**. *Debiasing Random Oblique Projections for Subsampled OLS and Fast CUR in High Dimensions*. May 2026. arXiv: 2605.24955 [math.NA]

# Subsampled OLS

Given $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{y} \in \mathbb{R}^n$,

$$\boldsymbol{\beta}_{\mathrm{OLS}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = \mathbf{X}^\dagger \mathbf{y}.$$

The classical subsampled solution is

$$\widetilde{\boldsymbol{\beta}} = (\mathbf{S}\mathbf{X})^\dagger \mathbf{S}\mathbf{y},$$

and the debiased solution is

$$\check{\boldsymbol{\beta}} = (\check{\mathbf{S}}\mathbf{X})^\dagger \check{\mathbf{S}}\mathbf{y}.$$

**Question**: how good statistically is the debiased $\check{\boldsymbol{\beta}}$ versus the classical solution $\widetilde{\boldsymbol{\beta}}$?

# Bias and variance around full OLS

Let

$$L(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2, \qquad \text{residue: } \mathbf{r} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}_{\mathrm{OLS}} = \mathbf{P}_{\perp}\mathbf{y}.$$

For a random estimator $\boldsymbol{\beta}$, define

- **Bias** as $\mathrm{Bias}_{\zeta}(\boldsymbol{\beta}) = L(\mathbb{E}_{\zeta}[\boldsymbol{\beta}]) - L(\boldsymbol{\beta}_{\mathrm{OLS}})$; and
- **Variance** as $\mathrm{Var}_{\zeta}(\boldsymbol{\beta}) = \mathbb{E}_{\zeta}[L(\boldsymbol{\beta})] - L(\boldsymbol{\beta}_{\mathrm{OLS}})$,

both measured in the original **full** least-squares objective.

# Classical subsampled OLS can be substantially biased

**Bias lower bound for classical subsampled OLS, [Niu+26]**

There exist $\mathbf{X}, \mathbf{y}$ and random sampling schemes such that, for every scalar correction $\gamma \in \mathbb{R}$,

$$\mathrm{Bias}_{\zeta}(\gamma\widetilde{\boldsymbol{\beta}}) = \Omega\left(\frac{d^2}{m^2}\right)\|\mathbf{r}\|^2.$$

- the bias is not a mere scalar inflation, and can be **nonuniform** across coordinates
- this (again) separates random sampling from (dense) random projection

# Debiased OLS: bias reduced without larger variance

For

$$\Delta(\mathbf{X}) = \mathbf{r}^\top \operatorname{diag}\{\ell_i(\mathbf{X})/(m\pi_i)\}_{i=1}^n \mathbf{r},$$

the debiased estimator satisfies

**Bias–variance characterization, [Niu+26]**

$$\operatorname{Bias}_\zeta(\check{\boldsymbol{\beta}}) = \varepsilon^2 \|\mathbf{r}\|^2, \qquad \operatorname{Var}_\zeta(\check{\boldsymbol{\beta}}) = \Delta(\mathbf{X}) + \varepsilon \|\mathbf{r}\|^2,$$

with

$$\varepsilon = \widetilde{O}\left(\sqrt{\frac{\theta_{\max}^3(\mathbf{X}) d^3}{m^3}}\right).$$

$\Rightarrow$Debiasing improves the bias while preserving the leading variance term.

# Special cases: leverage sampling and SRHT

## When debiasing matters

- **Uniform sampling**: can be biased if leverage scores are nonuniform.
- **General sampling**: matrix-valued and leverage-dependent debiasing.

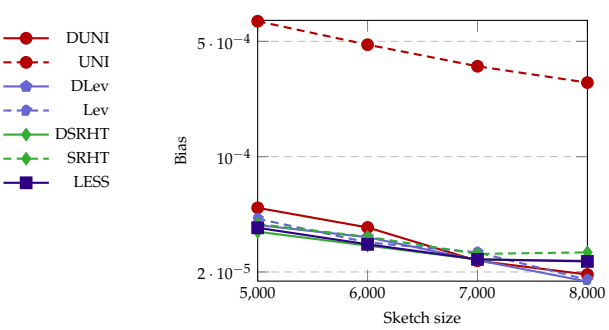
## When scalar correction is enough

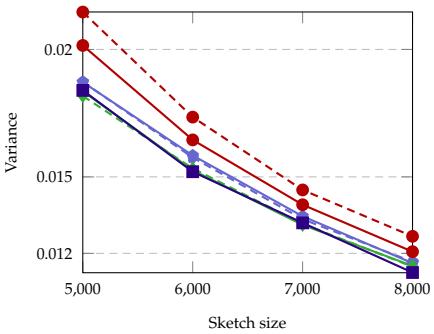
- **Exact leverage sampling**: the correction collapses to a scalar.
- **SRHT**: leverage scores are spread before uniform sampling.

## Conceptual "rule"

The more uniform the effective leverage profile, the closer random sampling behaves to (dense) random projection.

# Numerical evidence: debiasing reduces bias




- **Dashed** for classical sampling; and **Solid** for debiased
- Uniform sampling has the largest bias; debiasing brings it closer to leverage/SRHT/LESS
- Comparable variance after debiasing

# Take-away messages

1. **Unbiased sketches do not imply unbiased algorithms.**

$$\mathbb{E}[\mathbf{S}^\top \mathbf{S}] = \mathbf{I}_n \quad \not\Rightarrow \quad \mathbb{E}[(\mathbf{X}^\top \mathbf{S}^\top \mathbf{S}\mathbf{X})^{-1}] = (\mathbf{X}^\top \mathbf{X})^{-1}.$$

2. **(Inversion) bias of random sampling is generally matrix-valued.** Scalar debiasing works for projections and exact leverage sampling, but not for general sampling.
3. **Oblique projection is the right operator for subsampled OLS.** Debiasing $\widetilde{\mathbf{P}} = \mathbf{X}(\mathbf{S}\mathbf{X})^\dagger \mathbf{S}$ yields sharper bias–variance guarantees.
4. **RMT provides algorithmic corrections beyond asymptotic spectra.**

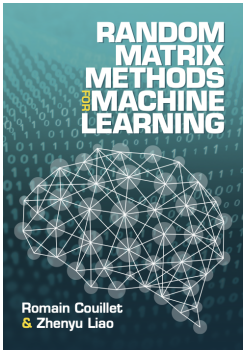
# References


- Michal Derezinski, Feynman T Liang, **Z. Liao**, and Michael W. Mahoney. "Precise expressions for random projections: Low-rank approximation and randomized Newton". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 18272–18283
- Michal Derezinski, **Z. Liao**, Edgar Dobriban, and Michael Mahoney. "Sparse sketches with small inversion bias". In: *Proceedings of Thirty Fourth Conference on Learning Theory*. Vol. 134. PMLR, 15–19 Aug 2021, pp. 1467–1510
- Chengmei Niu, **Z. Liao**, Zenan Ling, and Michael W. Mahoney. "Fundamental Bias in Inverting Random Sampling Matrices with Application to Sub-sampled Newton". In: *Proceedings of the 42nd International Conference on Machine Learning*. PMLR, Oct. 2025, pp. 46649–46692
- Chengmei Niu, Sachin Garg, Michał Dereziński, and **Z. Liao**. *Debiasing Random Oblique Projections for Subsampled OLS and Fast CUR in High Dimensions*. May 2026. arXiv: 2605.24955 [math.NA]

# One more thing

RMT for computer science and machine learning:

- **precise characterization** of average-case behavior (in RNLA and optimization)
- **improved novel methods** with performance guarantee!




- "*Random Matrix Methods for Machine Learning*" by Romain Couillet and **Z. Liao**, Cambridge University Press, 2022
- a pre-production version at `https://zhenyu-liao.github.io/book/`
- MATLAB and Python codes at `https://github.com/Zhenyu-LIAO/RMT4ML`

Thank you! Q & A?