Recent and Upcoming Developments in Randomized Numerical Linear Algebra for Machine Learning[*]

Michał Dereziński
University of Michigan
Ann Arbor, USA
derezin@umich.edu

Michael W. Mahoney
ICSI, LBNL, and University of California, Berkeley
Berkeley, USA
mmahoney@stat.berkeley.edu

ABSTRACT

Large matrices arise in many machine learning and data analysis applications, including as representations of datasets, graphs, model weights, and first and second-order derivatives. Randomized Numerical Linear Algebra (RandNLA) is an area which uses randomness to develop improved algorithms for ubiquitous matrix problems. The area has reached a certain level of maturity; but recent hardware trends, efforts to incorporate RandNLA algorithms into core numerical libraries, and advances in machine learning, statistics, and random matrix theory, have lead to new theoretical and practical challenges. This article provides a self-contained overview of RandNLA, in light of these developments.

CCS CONCEPTS

• **Computing methodologies → Machine learning algorithms**; • **Theory of computation → Design and analysis of algorithms**; • **Mathematics of computing → Probability and statistics**.

KEYWORDS

Matrix computations, Randomization, Optimization, Statistics

ACM Reference Format:
Michał Dereziński and Michael W. Mahoney. 2024. Recent and Upcoming Developments in Randomized Numerical Linear Algebra for Machine Learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3637528.3671461

1 INTRODUCTION

Matrices provide a natural structure with which to model data. For example, a matrix $A \in \mathbb{R}^{m \times n}$ can encode information about m objects, each of which is described by n features. Alternatively, a positive definite matrix $A \in \mathbb{R}^{n \times n}$ can encode correlations/similarities between all pairs of n objects. Motivated by large-scale data problems, recent years have witnessed many exciting developments in the theory and practice of matrix algorithms. Particularly remarkable is the use of randomization. Historically, in statistics, machine learning (ML), and domain sciences, randomization has been assumed to be a property of the input data, e.g., due to noise in the data generation mechanisms. In this more recent work on randomization, it is used as an algorithmic or computational resource.

Randomized Numerical Linear Algebra (RandNLA) is an interdisciplinary research area that exploits randomization as a computational resource to develop improved algorithms for large-scale linear algebra problems. From a *foundational perspective*, it has roots in theoretical computer science (TCS), deep connections with convex analysis, probability theory, and metric embedding theory, etc., as well as strong connections with scientific computing, signal processing, and numerical linear algebra (NLA). From an *implementational perspective*, well-engineered RandNLA algorithms beat highly-optimized software libraries for ubiquitous problems such as very over-determined least-squares, they scale well to parallel/distributed environments, and they beat state-of-the-art for a wide range of low-rank matrix approximation problems. From a *data analysis perspective*, RandNLA has strong connections with ML and statistics and many "non-methodological" applications of data analysis. More generally, of course, it is of continued importance since there is a growing interest in providing an *algorithmic and statistical foundation for modern large-scale data analysis.*

The area of RandNLA has achieved a certain level of maturity. As such, there are multiple reviews of the area from multiple different perspectives: introductory overviews (light on prerequisites) [29, 30]; broad and proof-heavy resources [60, 88, 89]; perspectives on interdisciplinary theory (light on proofs) [21, 59]; deep investigations of specific disciplinary topics [45, 50, 62, 63]; and approaches to high-quality software implementations [69]. Particularly notable is the current effort of incorporating RandNLA algorithms into the core numerical libraries (e.g., RandLAPACK and RandBLAS; see [69]) that lie at the foundation of virtually all computational tools in ML (and scientific computing and beyond).

This level of maturity, as well as recent demands by the ML community and recent trends in hardware, lead to new theoretical and practical challenges that did not exist a decade ago. For example: developing RandLAPACK and RandBLAS leads to new algorithmic and theoretical abstractions, different than those present in TCS or NLA, and different than those common in statistics and ML; recent developments in neural network training highlight important trade-offs between communication and computation, between forward accuracy and parameter stability, etc.; and recent developments in hardware have led to new trade-offs between latency and throughput, both at the model training and model inference stage. To complement these challenges, recent theoretical developments

[*]This article is an abridged conference version of the technical report [22]. For the full version, with additional material and details, see: https://arxiv.org/pdf/2406.11151. MD would like to acknowledge NSF CAREER for partial support. MWM would like to acknowledge the NSF and ONR via its BRC on RandNLA for partial support.

KDD '24, August 25–29, 2024, Barcelona, Spain
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671461

in Random Matrix Theory (RMT) have provided finer quantitative performance analysis than was possible with traditional RandNLA analysis tools. These theoretical developments come from RMT as well as from RandNLA itself, and they permit both finer analysis of existing methods as well as the possibility to develop novel methods that better bridge the theory-practice gap.

In this survey, we will provide a self-contained review/overview of RandNLA, in light of these recent developments, describing and highlighting upcoming and future trends. We will introduce the foundations of RandNLA and matrix sketching, with a particular focus on applications in ML and stochastic optimization, followed by an overview of recent developments in using sketching methods to gain stronger control on the convergence and generalization properties of stochastic algorithms. We will cover both the theoretical aspects of these techniques, as well as their applications in the context of important ML and data science research topics. Thus, our discussion will be relevant not only to theoretical researchers who wish to learn the latest advances in RandNLA, but also to a general audience of ML and data science researchers and practitioners, who want to incorporate RandNLA into large-scale data problems.

2 FOUNDATIONS OF "CLASSICAL" RANDNLA

In this section, we will describe the foundations of RandNLA theory, at least up until a few years ago. The basic idea is to construct a sketch (either data-aware or more commonly data-oblivious) that has parameters chosen basically to preserve the geometry of an entire low-dimensional subspace. This sketch can be interpreted in one of several complementary ways; and it can be used in one of several complementary ways. Understanding the details of these complementary approaches is crucial for understanding recent advances and upcoming developments in RandNLA for modern ML.

2.1 Matrix Multiplication

A core primitive in RandNLA is that of approximating the product of two matrices [27]. The basic problem is the following: given an $m \times n$ matrix A and an $n \times q$ matrix B, approximate the product $A \cdot B$. A key conceptual insight is that this problem can be expressed as approximating the sum of n rank-one matrices:

$$A \cdot B = \sum_{k=1}^{n} \left(A_{*k} \right) \cdot \left(\quad B_{k*} \quad \right).$$

Given this, a natural (random sampling) algorithm suggests itself:

```
1: Fix a set of probabilities $p_i$, $i = 1, \ldots, n$, summing up to 1.
2: **for** $t = 1, \ldots, s$ **do**
3:     Set $j_t = i$ according to $\mathbb{P}[j_t = i] = p_i$.
4: **end for** (Randomly pick $s$ terms of the sum according to $p_i$'s.)
5: Approximate $AB$ by summing the $s$ terms, after scaling.
```

If we let S be an $s \times n$ matrix whose tth row ($t = 1, \ldots, s$) has one non-zero, $S_{t,j_t} = 1/\sqrt{sp_{j_t}}$, then we can use a "sampling matrix" formalism [27] (a type of "sketching operator") to express this as:

$$AB \approx \frac{1}{s} \sum_{t=1}^{s} \frac{1}{p_{j_t}} a_{j_t} b_{j_t}^\top = (AS^\top) \cdot (SB) = C \cdot R,$$

where a_{j_t} and $b_{j_t}^\top$ are corresponding columns of A and rows of B. Now, if we use probabilities $p_i \propto \|a_i\|_2 \|b_i\|_2$ (row-norm sampling [27]), then we can minimize a very natural Frobenius norm error:

$$\mathbb{E}\left[\|AB - CR\|_F\right] = \mathbb{E}\left[\|AB - (AS^\top)(SB)\|_F\right] \le \tfrac{1}{\sqrt{s}} \|A\|_F \|B\|_F. \quad (1)$$

Here, by setting $s = 1/\epsilon^2$, we obtain a Frobenius norm error that is bounded above by ϵ, in the additive scale defined by $\|A\|_F \|B\|_F$.

This *Frobenius norm* bound given in (1) is used in *many* places in RandNLA. However, "better" *spectral norm* bounds of the form $\|AB - CR\|_2 \le \epsilon \|A\|_2 \|B\|_2$ are possible (after adjusting the sample size s; see [12]) via matrix versions of Chernoff/Bernstein concentration inequalities [88].

Within RandNLA, the main "use case" for approximate matrix multiplication arises when it is applied not directly to the matrices themselves, but rather to the "subspaces" that they define, giving rise to *subspace embeddings* [59, 89].

Definition 1. *Let A be an $n \times d$ matrix, and let U consist of orthonormal columns that are the basis of the column subspace of A. Then, an $s \times n$ matrix S is a subspace embedding for A if*

$$\|U^\top U - (SU)^\top SU\|_2 = \|I - (SU)^\top SU\|_2 \le \epsilon.$$

Here, for a sampling matrix S, the "correct" sampling probabilities p_i to use are proportional to the squared row norms $\|U_{i*}\|_2^2$, a.k.a. the *leverage scores* of A. These are expensive to compute exactly, but they are fast to approximate—explicitly or implicitly—in one of many ways [2, 11, 28, 65, 75].

Subspace embeddings are an extremely important concept in Classical RandNLA theory. They were introduced and first used implicitly in RandNLA by [31, 32], who relied on leverage score sampling [28], which can be viewed as a type of *data-aware sketching*. The first explicit definition of subspace embeddings was given by [33, 85], who focused on *data-oblivious* sketching; and these data-oblivious methods were popularized by [89]. Many constructions (random sampling and projection methods, deterministic constructions, hashing functions, etc.) satisfy this condition [89].

There are several complementary interpretations of a subspace embedding. Within NLA, it is an *acute perturbation* [86], meaning in particular that distances and angles are perturbed, but rank is not lost. Within TCS, it is a Euclidean space analogue of *Johnson-Lindenstrauss (JL) lemma* [64], meaning in particular that distances are approximately preserved for the infinite number of points on a unit ball in the low-dimensional space. More generally, the subspace embedding guarantee can be used to produce a *spectral approximation* of the matrix product $A^\top A$, in terms of the Loewner ordering of positive semi-definite (PSD) matrices, which provides a strong control on *all* the associated eigenvectors and eigenvalues. To be more precise, let A be an $m \times n$ matrix and $\epsilon \in [0, 1/2]$. If matrix S is an $\epsilon/2$ subspace embedding for the subspace defined by A; then $A^\top S^\top SA$ is a spectral approximation of $A^\top A$, i.e.,

$$\tfrac{1}{1+\epsilon} A^\top A \preceq A^\top S^\top SA \preceq (1+\epsilon) A^\top A, \quad (2)$$

where $\preceq$ is the PSD Loewner ordering. (We use $A^\top S^\top SA \approx_\epsilon A^\top A$ as a shorthand for (2) throughout.) The property $A^\top S^\top SA \approx_\epsilon A^\top A$ implies that any ith eigenvalue of the sketch $A^\top S^\top SA$ is an ϵ approximation of the corresponding eigenvalue of the data matrix $A^\top A$ (and similarly for the singular values of SA). It also extends to the control of various matrix functions of $A^\top A$, including the inverse, i.e., $(A^\top S^\top SA)^{-1} \approx_\epsilon (A^\top A)^{-1}$, square root, trace, quadratic form, and more.

This subspace embedding property is a "must must have" for the worst-case style of analysis provided by TCS. For everyone else, it provides a guiding principle, but (strictly speaking) it's optional. For example, if one is interested in providing good numerical implementations, then losing rank can still give a good preconditioner [3, 4]; and if one is interested in statistics and ML, then losing rank introduces a bit of bias that can be compensated for by greatly reducing variance [57, 58, 78]. *This leads to a theory-practice gap. As we will describe in Sec. 3, this is the key technical place where Classical RandNLA theory and Modern RandNLA theory differ, and why recent developments in Modern RandNLA theory open up the door for so many other use cases and applications of RandNLA ideas.*

2.2 Least-squares Approximation

We can use ideas from Sec. 2.1 for RandNLA methods for approximate matrix multiplication as a foundation to develop RandNLA methods for least-squares (LS) regression. Given $m \times n$ matrix A and m-dimensional vector b, the basic LS problem is to solve

$$x^* = \operatorname*{argmin}_{x \in \mathbb{R}^n} \|Ax - b\|_2 .$$

If $m \gg n$, then the problem is called overdetermined or overconstrained; and if $m \ll n$, then the problem is called underdetermined or underconstrained (or, especially in ML, overparameterized). In the overdetermined case, one can compute the solution in $O(mn^2)$ time (in RAM model), e.g., with normal equations, QR decompositions, or the Singular Value Decomposition (SVD). RandNLA gives faster algorithms for this ubiquitous problem, which is at the root of most other advances (including low-rank matrix approximation, as well as extensions to optimization problems, etc.) in the area.

Importantly, depending on the domain, "faster" can mean one of several different things: within TCS, "faster" is in terms of low-precision asymptotic worst-case theory; within NLA, "faster" is in terms of high-precision wall-clock time; and in large-scale implementations, one can compute (in Spark/MPI/etc.) low, medium, and high precision solutions on up to terabyte-sized data, and here "faster" depends mostly on communication.

Meta-algorithm for LS regression. Here is a meta-algorithm for solving LS regression problems with RandNLA methods.

```
1: Randomly sample a small number of constraints according to
   $A$'s leverage score sampling probabilities $\{p_i\}_{i=1}^m$.
2: Solve the LS regression problem on the resulting subproblem.
```

A *naïve version* of this meta-algorithm is *not promising*: it gives a $1+\epsilon$ relative-error approximation (since it approximates the solution using only a subset of constraints), that fails with probability δ (comparable to flipping a fair coin "heads," say, 30 times in a row), in roughly $O(mn^2)$ time, i.e., as long as the exact deterministic method (since, as stated, it computes the leverage scores exactly).

On the other hand, a *non-naïve version* of this meta-algorithm is *very promising*: it gives the best worst-case algorithm in RAM [28, 33, 85] (using Sketch-and-Solve, described below); it beats LAPACK for high precision in wall-clock time [3, 66, 81] (using Sketch-and-Precondition, described below); it leads to super-terabyte-scale implementations in parallel/distributed environments [38, 92]; and it gives the foundation for low-rank approximations and the rest of RandNLA [30–32, 59].

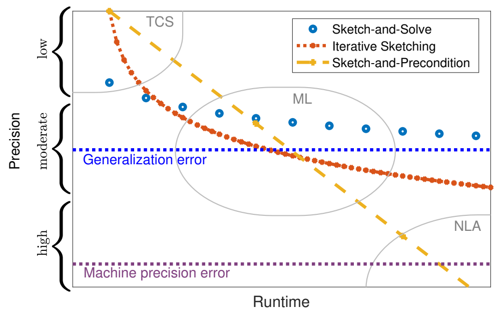

Figure 1: Three algorithmic paradigms for RandNLA methods: "Classical" (subspace embedding based) RandNLA theory is most appropriate for low precision (with Sketch-and-Solve) and high precision (with Sketch-and-Precondition); while "Modern" (RMT-based) RandNLA theory is well-suited for moderate precision (with Iterative Sketching), which is of increasing interest in modern ML applications.

Fundamental structural result. For LS approximation, here is the fundamental structural result that is at the heart of RandNLA-based methods [33, 59]. (This is a deterministic result that holds for any matrix S, and it is central to RandNLA theory by choosing S to be an appropriate sketching operator.) Consider the "sketched" LS approximation problem:

$$\tilde{x} = \operatorname{argmin}_x \|S(Ax - b)\|_2,$$

where S is *any* matrix. If S satisfies the two basic conditions:

$$1)\ \sigma^2_{\min}(SU_A) \geq 1/\sqrt{2} \quad \text{and} \quad 2)\ \|U_A^\top S^\top S b^\perp\|_2 \leq \sqrt{\epsilon/2}\,\|Ax^* - b\|_2,$$

where $b^\perp = b - U_A U_A^\top A$ and U_A is the orthonormal basis of A, then:

$$\|A\tilde{x} - b\|_2 \quad \leq \quad (1+\epsilon)\|Ax^* - b\|_2 .$$

Importantly, each of the two basic conditions can be viewed as approximate matrix-matrix multiplication [30, 59].

RandNLA provides a plethora of sketching methods (random sampling and projection methods, deterministic constructions, hashing functions, etc.) for S so as to satisfy these two basic conditions. As described in the many reviews of RandNLA, the choice of which method one uses typically depends on one's goals: to provide TCS-style worst-case theory; to provide NLA-style implementations; to use in large-scale and/or ML settings; etc.

Three algorithmic paradigms for RandNLA methods. If we want to use RandNLA methods more generally, then there are three general paradigms—that apply both to LS methods as well as more broadly than LS (e.g., to low-rank matrix approximation).

(1) *Sketch-and-Solve*: Here [28, 33, 85], we randomly construct a *smaller* LS problem, and then solve it using a traditional NLA method. This is the simplest approach to highlight the structure of RandNLA theory; it is most convenient for TCS-style theory; it leads to low-precision estimates, e.g., $\epsilon = 0.1$.
(2) *Iterative Sketching*: Here [43, 76, 91], we repeatedly sketch or sub-sample the problem randomly, and then iteratively refine the estimate using a convex optimization method. This includes Preconditioned Weighted SGD, Sketch-and-Project, and Newton Sketch; it is the most convenient for ML-style stochastic optimization; and it leads to moderate-precision estimates, e.g., $\epsilon = 10^{-3}$.

(3) *Sketch-and-Precondition*: Here [3, 66, 81], we randomly construct an *equivalent* but well-conditioned problem, and then solve it using a traditional deterministic NLA iterative method. This is the best (usually) for high-quality numerical solutions; it is most convenient for NLA-style implementations; and it leads to high-precision solutions, e.g., $\epsilon = 10^{-10}$.

See Fig. 1 for an illustration of these three algorithmic paradigms. The first and the third have received the greatest amount of attention within RandNLA, in part due to the (TCS and NLA based) style of theory used in Classical RandNLA. The new theoretical developments in Modern RandNLA (described in Sec. 3) are particularly well-suited to the second (and for ML based theory).

2.3 Low-rank approximation

We can use the ideas from Sec. 2.2 to develop RandNLA methods for low-rank (LR) approximation. Many details change, e.g., for LS problems the matrix is usually tall, while for LR problems both dimensions may be large, in which case there is a rank parameter. This is described in more detail in the full technical report [22].

3 FOUNDATIONS OF "MODERN" RANDNLA

In this section, we will describe relatively-recent work that provides a foundation for current and upcoming developments in RandNLA. One surprising aspect of Classical RandNLA theory, as outlined in Sec. 2, is that there is actually very little use of randomness. Essentially, all the randomness is "filtered through" the subspace embedding. That is sufficient for many TCS and NLA style objectives (although with Sketch-and-Solve and Sketch-and-Precondition, the two areas use that embedding differently), but it is often "overkill" when RandNLA methods are used in broader (statistical, ML, optimization) pipelines (that typically use some form of Iterative Sketching). This has applications, in particular, in stochastic optimization and traditional statistical resampling methods, where the JL-style guarantees that come with a subspace embedding are sufficient but not necessary. Modern RandNLA theory basically asks for a stronger form of "Algorithmic Gaussianization" than the JL Lemma provides, in the sense that sketches look more Gaussian-like than what is provided with pair-wise JL-like guarantees. To accomplish this, we must make connections with ideas from non-asymptotic RMT, which have proven fruitful across data science and ML.

3.1 Algorithmic Gaussianization via RMT and LESS Embeddings

To understand the basic issue, let us revisit the subspace embedding, where for an $n \times d$ orthogonal matrix U (a column basis matrix of A) we seek a matrix $S \in \mathbb{R}^{s\times n}$ such that $(SU)^\top SU \approx_\epsilon U^\top U = I$. Here, s must be larger than d; but how much larger does it need to be?

The proportional regime. Let us consider the extreme case (which is in fact the most common) where $s \approx d$ to within a constant factor, and suppose that S is a scaled i.i.d. Gaussian sketching matrix. Thanks to the rotation invariance of Gaussian distribution, SU is also Gaussian, and so we can obtain (up to lower order terms):

$$\sigma_{\min}(SU) \approx 1 - \sqrt{\frac{d}{s}}, \qquad \sigma_{\max}(SU) \approx 1 + \sqrt{\frac{d}{s}}. \tag{3}$$

This follows from the Marchenko-Pastur (MP) law [5]. Such a Gaussian sketch not only gives us sharp control on subspace embeddings, but also on most Classical RandNLA tasks, e.g.:

- Sketch-and-Solve: If we solve $\tilde{x} = \mathrm{argmin}_x \|S(Ax-b)\|_2^2$, then $\mathbb{E}\|A(\tilde{x}-x^*)\|_2^2 = \frac{d}{s-d-1}\|Ax^*-b\|_2^2$ for $s \geq d+2$.
- Sketch-and-Precondition: If we construct R^{-1} from the QR of SA, then $\mathrm{cond}(AR^{-1}) \leq 6$, with high probability for $s \geq 2d$.
- LR approximation: If we compute $Q = \mathtt{orth}(AS)$, then $\mathbb{E}\|A - QQ^\top A\|_F^2 \leq \left(1+\frac{k}{s-k-1}\right)\cdot\|A-A_k\|_F^2$ for $s \geq k+2$.

From these results, traditional RandNLA-style guarantees (summarized in Sec. 2) are straightforward to derive. These results are all relatively easy to show for i.i.d. Gaussian matrices; and results of this form are common in NLA and scientific computing applications of RandNLA [45]. The basic question is whether we can obtain similar results with (faster) non-Gaussian sketches.

Inversion bias. To answer this question, we need a notion of sketch quality (akin to subspace embedding) that can capture the "Gaussian-like" behavior—*in the proportional regime*. Working in the proportional regime, where the aspect ratio of the matrix is constant, is key here—it is much more realistic, and if the aspect ratio is not constant then JL-like results are straightforward to derive. One way to capture this Gaussian-like behavior is through *inversion bias* [20, 37]. The basic idea should be obvious: for a real-valued random variable X, we know that $\mathbb{E}[X^{-1}] \neq \left(\mathbb{E}[X]\right)^{-1}$. A similar phenomenon occurs for random sketch matrices, $\tilde{A} = SA$:

$$\mathbb{E}[(\tilde{A}^\top\tilde{A})^{-1}] \neq (A^\top A)^{-1}, \quad \text{even though} \quad \mathbb{E}[\tilde{A}^\top\tilde{A}] = A^\top A. \tag{4}$$

This bias is especially pronounced in the proportional regime, due to the random singular value fluctuations described by the MP law.

Why focus on the inverse? Let us for a moment consider a statistical perspective on sketching. Here, the sketched covariance matrix $A^\top S^\top SA$ can be viewed as a *sample covariance estimator* of the "population covariance matrix" $A^\top A \in \mathbb{R}^{d\times d}$. A natural question is: how does the spectrum differ between the *sample* and the *population* covariance? RMT answers this by looking at the *resolvent matrix*:

$$(A^\top S^\top SA - zI)^{-1} \quad \text{for} \quad z \in \mathbb{C}\setminus\mathbb{R}_+.$$

The Stieltjes transform (normalized trace of the resolvent, central to RMT) exhibits an *inversion bias*, leading to a discrepancy between the sample and population. In traditional RMT, controlling this inversion bias lets us characterize the limiting eigenvalue distribution as $s, n, d \to \infty$. Recent works [17, 19, 20] have extended this RMT analysis to the *non-asymptotic* RandNLA theory, essentially showing that: *If we can correct the inversion bias of the sketch $\tilde{A} = SA$, then we can recover stronger Gaussian-like sketching performance for faster sketches across RandNLA tasks.*

Correcting the bias (for Gaussian sketches). When $\tilde{A} = SA$ is an $s \times d$ Gaussian sketch, scaled so that $\mathbb{E}[\tilde{A}^\top\tilde{A}] = A^\top A$, then there is a very simple correction for the inversion bias:

$$\mathbb{E}\left[(\gamma\tilde{A}^\top\tilde{A})^{-1}\right] = (A^\top A)^{-1} \quad \text{for} \quad \gamma = \frac{s}{s-d-1}.$$

This simple fix does *not* hold for other (faster) sketching methods that have been used in RandNLA, e.g., Hadamard-based projections, sub-Gaussian sketches, very sparse sketches, sampling methods, etc. The basic reason is that other sketches are *not* perfectly rotationally symmetric. Thus, among other things, they could lose rank (with very small probability) and/or suffer from "coupon collector" problems. In general, for faster sketching methods, the *inversion bias occurs differently in each direction*, and so we cannot correct it with a single dimensional rescaling factor.

Nearly-unbiased sketches. Going beyond Gaussian sketches motivates the notion of a nearly-unbiased estimator [20].

Definition 2. *A random PSD matrix $\tilde{C}$ is an* (ϵ, δ)-unbiased estimator *of C if there is an event $\mathcal{E}$ that holds with probability* $1-\delta$ *such that, when conditioned on $\mathcal{E}$,*

$$\mathbb{E}_{\mathcal{E}}[\tilde{C}] \approx_\epsilon C \quad \text{and} \quad \tilde{C} \preceq O(1)\cdot C.$$

Let S be an $s\times n$ random matrix such that $\sqrt{s}\,S$ has i.i.d. $O(1)$-sub-Gaussian entries with mean zero and unit variance (i.e., a direct extension of a Gaussian sketch). Building on the Stieltjes transform analysis, it is possible to show that for this sketching matrix we can nearly correct inversion bias, in the sense that $(\frac{s}{s-d}A^\top S^\top SA)^{-1}$ is an (ϵ,δ)-unbiased estimator of $(A^\top A)^{-1}$ for $\epsilon = O(\frac{\sqrt{d}}{s})$ and $\delta = e^{-cs}$ (Proposition 4 in [20]). In other words, conditioned on a high probability event $\mathcal{E}$, we have

$$\mathbb{E}_{\mathcal{E}}\left[(\tfrac{s}{s-d}A^\top S^\top SA)^{-1}\right] \approx_\epsilon (A^\top A)^{-1}, \quad \text{for} \quad \epsilon = O(\tfrac{\sqrt{d}}{s}). \qquad (5)$$

Let us compare/contrast this notion of an (ϵ,δ)-unbiased estimator with JL / subspace embeddings in Def. 1. For a subspace embedding, we have that

$$\text{Subspace embedding:} \quad (A^\top S^\top SA)^{-1} \approx_\eta (A^\top A)^{-1}, \quad \eta = \Theta\Big(\sqrt{\tfrac{d}{s}}\Big).$$

Thus, we see that the *average-case* analysis in (5) is sharper than what is possible to recover with the *high-probability* subspace embedding analysis, by at least a $\sqrt{s}$ factor.

Averaging RandNLA estimators. Implications of this Modern RMT-style analysis for RandNLA are discussed in later sections. However, as an immediate consequence, we can establish various "model averaging" schemes for RandNLA-based estimators, that can be used in statistical/ML/optimization pipelines.

The premise behind model averaging is that if we can produce a bias-corrected estimator $\tilde{x}$, i.e., $\mathrm{Bias}^2(\tilde{x}) \ll \mathrm{Var}(\tilde{x})$, then we can boost its accuracy by averaging multiple independent copies of this estimator. When the estimator relies on some linear functional of the inverse matrix $(A^\top S^\top SA)^{-1}$, as in sketched LS [37] and a number of other RandNLA algorithms [17], then this follows (with some small additional effort) from (5). For example, for q independent sub-Gaussian sketches S_i of size $s \geq O(d+q)$, if we average the LS estimates $\tilde{x}_i = \mathrm{argmin}_x \|S_i(Ax-b)\|_2$, then the aggregate estimator $\bar{x} = \frac{1}{q}\sum_{i=1}^q \tilde{x}_i$ satisfies [37]:

$$\mathbb{E}\|A\bar{x} - b\|_2 \leq (1+\epsilon)\|Ax^* - b\|_2, \quad \text{for} \quad \epsilon = O\big(\tfrac{d}{qs}\big).$$

Extending RMT-style analysis to fast sketching. Compared to classical JL or subspace embedding approaches, most RMT for sketching requires different "Gaussianization" assumptions and different parameter regimes (e.g., the proportional regime). Most out-of-the-box theory applies only to (expensive) dense Gaussian or (still expensive) dense sub-Gaussian sketching matrices. Given the widespread interest in fast sketching methods, the question is: *Can we extend this line of work to fast sketches, e.g., sparse or structured?*

Several recent results have provided an affirmative answer to this question [10, 15, 20, 26, 52], including: non-asymptotic RMT-style analysis for sparse sketches based on Leverage Score Sparsification (LESS) [10, 15, 20] and asymptotic RMT-style analysis for structured sketches based on randomized Hadamard transforms [26, 52]. These results essentially show that certain fast sketching operators are "close" in a strong enough sense to sub-Gaussian matrices, typically by relying on two structural conditions which are needed for S to ensure small inversion bias:

(1) *Subspace embedding*: this is the standard guarantee from Def. 1; using Modern RMT [9], sharp MP-style subspace embedding guarantees as in (3) can be recovered for fast sketching operators such as LESS [10].
(2) *Restricted Bai-Silverstein inequality*: this is the key novelty that provides a variance bound for random quadratic forms [20]; and it is related to the Hanson-Wright inequality [84].

Based on these ideas, we can reduce the cost of applying sub-Gaussian sketches down from $O(nds)$, while still recovering (5), i.e., we can construct more efficient sketches for which it is possible to correct inversion bias and recover other Gaussian RandNLA guarantees. For example, given a tall $n\times d$ matrix A, we can compute an $s\times d$ sketch SA (LESS embedding) in near-linear time $\tilde{O}(nd + sd^2)$ such that $(\frac{s}{s-d}A^\top S^\top SA)^{-1}$ is an (ϵ,δ)-unbiased estimator of $(A^\top A)^{-1}$ with $\epsilon = O(\frac{\sqrt{d}}{s})$, matching the guarantee for sub-Gaussian sketches (Theorem 8 in [20]).

3.2 RMT for Sampling via DPPs

The astute reader may wonder whether the developments described in Sec. 3.1 that hold for data-oblivious sketching methods also hold for data-aware sampling methods. After all, they are based on RMT-based sketching in the proportional regime, and in some sense sampling is inherently "non-RMT." For example, it involves coordinate axes and coordinate-aligned subspaces, and lower bounds arise that are due to the Coupon Collector problem. Yet, using Determinantal Point Processes (DPPs), one can show that very similar RMT results hold for certain data-aware sampling methods [21]. This is described in more details in the full technical report [22].

4 ADVANCES IN RANDNLA FOR OPTIMIZATION

In stochastic optimization, the goal is to minimize (or maximize) an objective function, when one or more of the input parameters is subject to randomness. It arises in many areas, most prominently with stochastic gradient descent (SGD) based methods for training ML models [8]. Leveraging Modern RandNLA techniques and theory (from Sec. 3) within the framework of stochastic optimization leads to the Iterative Sketching paradigm (of Fig. 1). This has manifested itself in many ways, including: PW-SGD [91], a variant of SGD which is preconditioned and weighted using RandNLA techniques; Subsampled Newton [35, 82, 90] and Newton Sketch [77], which apply the Sketch-and-Solve paradigm to Hessian estimation; and Sketch-and-Project [24, 25, 43], a modern extension of the classical Kaczmarz method [49], which uses sketching to achieve *implicit preconditioning* in stochastic methods. Recent advances in RandNLA (described in Sec. 3.1) have proven fruitful in obtaining theory for stochastic optimization algorithms based on Iterative Sketching, which comes with much smaller theory-practice gap compared to traditional approaches.

Consider the following standard finite-sum optimization task:

$$\text{Minimize} \qquad f(x) = \frac{1}{n}\sum_{i=1}^{n} \psi_i(x), \tag{6}$$

where x is a parameter (or weight) vector, and where each $\psi_i(x)$ corresponds to the loss for a single data point. In optimization, we generally approach this problem by iteratively refining a sequence of estimates $x_0, x_1, x_2, \ldots$, relying on local information about the objective function. We can divide optimization methods into *first-order* methods, which use gradient (first derivative) information of f at x_t, e.g., gradient descent; and *second-order* methods, which additionally rely on Hessian (second derivative) information.

Although introducing randomization in optimization algorithms has proven very effective in ML, with variants of SGD (AdaGrad, Adam, SVRG, etc.) being widely-used in practice, SGD-based methods still often suffer from instability, large variances, and extreme sensitivity to hyperparameter choice [48, 94]. In this context, randomized algorithms based on Modern RandNLA theory can be used to provide better stability, to inject better curvature information, and to reduce communication and computation costs.

4.1 Gradient Sketch

With SGD, we estimate the first-order (gradient) information by subsampling the components ψ_i of the objective, and using those to construct an unbiased estimate $\hat{g}_t$:

$$x_{t+1} = x_t - \eta_t \hat{g}_t, \qquad \text{where} \quad \mathbb{E}[\hat{g}_t] = \nabla f(x_t).$$

Despite its popularity, SGD has a number of limitations, e.g., large variance $\mathbb{E}\big[\|\hat{g}_t - g_t\|^2\big]$, which slows the convergence near the optimum, as well as sensitivity to hyper-parameters (such as step size, mini-batch size, etc.). RandNLA offers a number of techniques for addressing these limitations, such as using weighted gradient sampling based on leverage scores or other importance scores, as well as using a sketching-based preconditioner.

Preconditioned Weighted SGD. Consider the Preconditioned Weighted SGD (PW-SGD) method (which is shown here for the LS problem, $f(x) = \frac{1}{n}\|Ax - b\|^2$, but which is applicable far beyond this setting [1, 14, 34, 39]):

```
1: Compute $SA$ with some sketching operator $S$
2: Compute $R$ such that $SA = QR^{-1}$ for orthogonal $Q$
3: Compute leverage score estimates $\tilde{l}_i$ for $A$
4: for $t = 0$ to $T - 1$ do
5:     Compute $g_t \leftarrow \frac{1}{s}\sum_{i=1}^{s} \frac{Z}{\tilde{l}_{I_i}} \nabla\psi_{I_i}(x_t)$,   $\Pr(I_i) \propto \tilde{l}_{I_i}$.
6:     Compute $x_{t+1} \leftarrow x_t - \eta_t RR^\top g_t$
7: end for
```

This algorithm uses a sketching operator S to construct the R, which is the preconditioner of the problem, since its spectrum approximates the spectrum of (the inverse of) A. Moreover, the algorithm uses leverage score estimates of A, which are used for subsampling the gradients (but this can be replaced by other sketching/sub-sampling schemes).

Thanks to a combination of preconditioning and importance sampling, this version of PW-SGD can completely avoid any condition number dependence in its convergence rate [10]. In particular, suppose that the *leverage score* estimates satisfy: $\tilde{l}_i \geq l_i(A)/\alpha$ for all i. Then, letting $\eta_t := \frac{\beta}{1+\beta t/8}$ for $\beta = \frac{k/8}{k+4\alpha d}$, this version of PW-SGD satisfies:

$$\mathbb{E}[f(x_t) - f(x^*)] \leq \frac{f(x_0)}{1 + st/(c\alpha d)} \qquad \forall_{t\geq 1}.$$

The key advantage here is that the resulting iteration complexity $t = O(d/s\epsilon)$ is entirely independent of the number of data points n or of the condition number κ of matrix A [10], whereas classical SGD may require as many as $O(n\kappa^2/s\epsilon)$ iterations. For reaching a *moderate precision* solution (the regime of greatest interest in ML and data science applications), the computational cost comparison shows that this method is faster than Sketch-and-Solve or Sketch-and-Precondition [10].

Other RandNLA-based approaches for sketching the gradient information of the objective include: in non-finite-sum settings, e.g., Sega [47]; for distributed/federated learning, e.g., FetchSGD [83]; methods inspired by randomized coordinate descent [54, 74]; randomized preconditioning for other stochastic gradient methods, e.g., Preconditioned SVRG and SVRN [14, 39], and more [42, 56].

4.2 Hessian Sketch

RandNLA methods have proven particularly effective at efficiently extracting second-order information about the optimization objective [17, 77, 82]. This has led to many sketching-based Newton-type optimization algorithms. Recall that Newton's method represents the paradigmic second-order optimization algorithm, which minimizes a local quadratic approximation of the objective using gradient $g_t = \nabla f(x_t)$ and Hessian $H_t = \nabla^2 f(x_t)$:

$$x_{t+1} = x_t + \operatorname*{argmin}_v \left\{ g_t^\top v + \frac{\eta_t}{2} v^\top H_t v \right\} = x_t - \eta_t H_t^{-1} g_t.$$

As an example, consider a standard Generalized Linear Model (GLM), $f(x) = \frac{1}{n}\sum_{i=1}^{n} l_i(a_i^\top x) + \frac{\gamma}{2}\|x\|^2$, where $a_i^\top x$ represents a linear prediction associated with a data point $a_i \in \mathbb{R}^d$, and loss l_i encodes the prediction error, dependent on a label y_i. For instance, in logistic regression, we have $y_i = \pm 1$ and $l_i(a_i^\top x) = \log(1 + e^{-y_i a_i^\top x})$. Here, the Hessian at x_t is given by:

$$\nabla^2 f(x_t) = \tfrac{1}{n} A^\top D_t A + \gamma I, \qquad D_t := \operatorname{diag}(l_1''(a_1^\top x_t), \ldots, l_n''(a_n^\top x_t)),$$

where the dominant cost is the $O(nd^2)$ matrix multiplication $A^\top D_t A$. We can reduce this cost with RandNLA sketching or sub-sampling, by repacing $A^\top D_t A$ with $\tilde{A}_t^\top \tilde{A}_t$ where $\tilde{A}_t = S_t D_t^{1/2} A$ for some sketching matrix S_t. One version of this is the Newton Sketch [77]:

$$x_{t+1} = x_t - \eta_t \hat{H}_t^{-1} g_t, \quad \hat{H}_t = \tfrac{1}{n}\tilde{A}_t^\top \tilde{A}_t + \gamma I.$$

RandNLA guarantees such as the subspace embedding are sufficient (but not necessary) to ensure that $\hat{H}_t$ provides a good enough approximation to enable accelerated local convergence in time $\tilde{O}(nd)$.

These approaches have also been extended to distributed settings via RMT-based model averaging, with applications in ensemble methods, distributed optimization, and federated learning [17, 44, 52, 53, 71]. Further RandNLA-based Newton-type methods include: Subsampled Newton [6, 7, 35, 82]; Hessian approximations via randomized Taylor expansion [1] and low-rank approximation [23, 36]; Hessian diagonal/trace estimates via Hutchinson's method [67] and Stochastic Lanczos Quadrature, particularly for non-convex problems, e.g., PyHessian [93], AdaHessian [94]; and finally Stochastic Quasi-Newton type methods [51, 68].

4.3 Sketch-and-Project

The Sketch-and-Project framework has gained attention as a powerful methodology within the Iterative Sketching paradigm of RandNLA. While this framework has found applications across the stochastic optimization landscape, it originally arose from randomized algorithms for solving systems of linear equations.

Solving an $m \times n$ linear system $Ax = b$ can be viewed as an instance of the LS problem, i.e., minimizing the objective $f(x) = \|Ax-b\|_2^2$. As discussed in Sec. 2.2, Classical RandNLA has addressed this problem in the *highly* over- or under-determined settings (i.e., when A is very tall or wide). However, in many applications, particularly high-dimensional settings that arise in ML, m and n are (equal or) of comparable size.

For these nearly-square matrix problems, randomization can still be beneficial, in particular when using Modern RandNLA methods. For instance, consider the classical Kaczmarz algorithm, which solves a linear system of m equations $a_i^\top x = b_i$ via the following iterative procedure, starting from some x_0:

```
1: for $t = 1, 2, \dots$ do
2:     Select index $I_t \in \{1, ..., m\}$
3:     $x_{t+1} \leftarrow$ Project $x_t$ onto the solutions of equation $a_{I_t}^\top x = b_{I_t}$
4: end for
```

This simple procedure has been known for a long time [49], but only with the use of randomization are we able to characterize its convergence [87]: if the Kaczmarz method selects index I_t randomly, with probability proportional to $\|a_{I_t}\|_2^2$, then:

$$\mathbb{E}\,\|x_t - x^*\|_2^2 \le \left(1 - \frac{\sigma_{\min}^2(A)}{\|A\|_F^2}\right)^t \|x_0 - x^*\|_2^2.$$

We can interpret this Randomized Kaczmarz (RK) method as a Weighted SGD algorithm solving the finite-sum minimization problem (6) with $\psi_i(x) = (a_i^\top x - b_i)^2$ [74], drawing a parallel with PW-SGD as described in Sec. 4.1.

However, this paradigm becomes quite different from PW-SGD once we select more than one equation at a time, giving rise to Sketch-and-Project [43]: *Sample a random $k \times m$ matrix $S = S(t)$, and project x_t onto the solutions of $SAx = Sb$:*

$$x_{t+1} = \operatorname*{argmin}_x \|x_t - x\|_2^2 \quad \text{subject to} \quad SAx = Sb.$$

We recover RK if the matrix S is chosen to be the indicator vector of the equation I_t. However, this general framework (with Modern RandNLA tools from Sec. 3) allows for other natural choices, such as selecting blocks of equations (Block Kaczmarz, e.g., [72, 73, 79]) or sketching the input matrix A using any of the methods described earlier. The Sketch-and-Project framework has been used to capture and extend other stochastic optimization methods, including Coordinate Descent [40, 54, 55], and to develop more general purpose first- and second-order optimization algorithms [41, 46, 51, 95].

Going beyond RK, the convergence analysis of Sketch-and-Project has proven much more challenging, largely due to the complex interdependence between the distribution of the sketching matrix S and the spectrum of the input matrix A. However, by using tools from Modern RandNLA theory [19, 70, 80], we can relate this convergence to the quality of the sketch SA as a low-rank approximation of the data [24]:

$$\mathbb{E}\,\|x_t - x^*\|_2^2 \lesssim \left(1 - \frac{k\sigma_{\min}^2(A)}{\mathbb{E}\|A - AP_{SA}\|_F^2}\right)^t \|x_0 - x^*\|_2^2. \qquad (7)$$

Here, $\mathbb{E}\|A - AP_{SA}\|_F^2$ is the projection error of the low-rank approximation of A produced from the sketch SA. This analysis has revealed new classes of problems (often overlapping with ML domains that exhibit low-rank structure) for which Sketch-and-Project methods offer dramatically improved peformance over the more traditional Sketch-and-Precondition paradigm [18, 25], thanks to the *implicit preconditioning* phenomenon that is described by (7).

5 ADVANCES IN RANDNLA FOR ML / STATS

So far, we have primarily focused on how RandNLA can be used to address ML and data science problems from a purely algorithmic perspective: the data (e.g., matrix A and vector b) is given and deterministic, and our goal is to directly compute or estimate a property of the data (e.g., the LS solution). With this problem formulation, randomization has entered solely from the algorithm design perspective, and thus it is fully controllable by the practitioner.

However, when we consider the same problems from a statistical perspective, the data is often assumed to be random itself, e.g., coming from some data distribution (which may or may not be known) and distorted by some random noise (e.g., Gaussian). In this case, our goal is often to estimate a quantity that may depend on an unobserved part of the data distribution, in which case we must account for the *generalization error*, in addition to the *approximation error* that was our focus so far. In this section, we delve into how randomization coming from RandNLA interacts with the inherent data randomness, and how that affects the design and implementation of these methods.

5.1 Statistical Learning Approaches

One approach is to adopt a statistical learning theory perspective on generalization error, which is most traditionally exemplified by the probably approximately correct (PAC) model of learning. Here, we often assume very little about the underlying data distribution, leading to a worst-case view of the generalization error. This forces our algorithmic approaches to exhibit a certain degree of robustness to the unobserved data. RandNLA techniques naturally inject such robustness into ML algorithms, as they are also designed to work well under worst-case settings.

Robust semi-supervised learning and *kernel-based learning* provide two examples of using RandNLA methods with this approach. This is described in more details in the full technical report [22].

5.2 Statistical Inference Approaches

Another approach is to consider how RandNLA interacts with and informs statistical inference approaches to modeling and analyzing data. In statistical inference, it is common to impose strict assumptions on the generative model of the data, e.g., that the feature vectors are coming from a Gaussian or sub-Gaussian distribution, and that the predicted variable follows some underlying linear model distorted by noise. This generative modeling approach enables a wide range of inferential tools for designing, optimizing and evaluating estimators. These tools include cross-validation, feature selection, reliable confidence intervals, the Bootstrap method, the Jackknife, etc. This might seem in stark contrast to the RandNLA approach to data, which has largely been centered on robustness

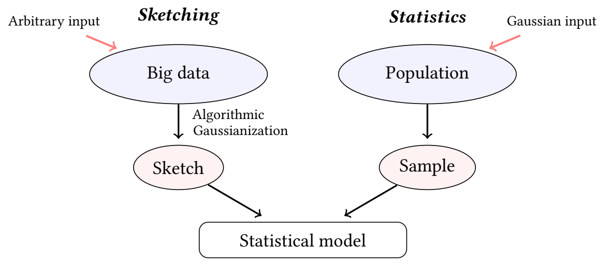

Figure 2: Connection between RandNLA and statistical inference. Even though the input matrix A for a RandNLA algorithm is often arbitrary and deterministic, the Algorithmic Gaussianization effect of sketching turns this matrix into a random data sample $\tilde{A}$ (the sketch), which follows a generative model like those in statistical inference.

to the worst-case and avoiding data assumptions. As we have seen, a typical framework for RandNLA algorithms is that we are given an arbitrary input matrix A with no distributional assumptions.

To see that there is a strong connection, however, recall that instead of operating directly on A, we first compute a smaller sketch $\tilde{A} = SA$, and then we use that sketch to estimate the target properties of the data. *This means that, even though the input matrix A is deterministic and arbitrary, the matrix $\tilde{A}$ which we use for the data analysis follows a well-defined generative model—and one which, for many sketching operators, is very close to a sub-Gaussian data model used in statistical inference.* See Fig. 2 for an illustration. This motivates a statistical view of algorithms based on sketching, unlocking the vast wealth of inferential tools that can be applied to any approaches based on the Sketch-and-Solve paradigm and beyond.

Bootstrap error estimation for sketching is an example of this statistical view, and it can also be used to provide a *separation between sketching methods*. This is described in more details in the full technical report [22].

5.3 Connections with Modern RMT

One of the key tools in Modern RandNLA for developing fine-grained understanding of the performance of RandNLA algorithms, including those inspired by the above statistical inference viewpoint, has been the use of techniques from both asymptotic and non-asymptotic RMT. These techniques are particularly important when dealing with matrices with a statistically well-understood data distribution, where worst-case performance guarantees can be significantly misleading.

While Classical RandNLA has focused on describing the performance of the algorithms solely in terms of the dimensions of the input (or possibly also its sparsity), it has been shown that for many tasks it is the spectral decay profile of the data (i.e, the rate of descrease of the singular values of a matrix A) that is most informative about the performance of these algorithms. In fact, for many ML tasks, the spectral decay profile of the data can often be sharply characterized, for example, in terms of whether we expect a heavy-tailed, short-tailed, or spiked profile [61]. This can have a dramatic effect on the design and performance of RandNLA algorithms.

These tools can be used to characterize *multiple-descent in LR approximation* and *implicit regularization in sketching*. This is described in more details in the full technical report [22].

6 PUTTING RANDOMNESS INTO NEXT-GENERATION SOFTWARE

One of the promises of RandNLA has been that it will lead to improved implementations in practice. Recent years have seen the convergence of several unprecedented changes, including formidable new system design constraints and revolutionary levels of hardware heterogeneity; and, due to these changes, much of the essential software infrastructure of computational science and engineering is, or will soon be, obsolete. These challenges motivated the BALLISTIC ("Basic ALgebra LIbraries for Sustainable Technology with Interdisciplinary Collaboration") project [13], which aims to enhance and update BLAS, LAPACK, and ScaLAPACK ("Scalable Linear Algebra PACKage") by "incorporating them into a layered package of software components ... that provides users seamless access to state-of-the-art solver implementations through familiar and improved Sca/LAPACK interfaces." As part of the BALLISTIC project [13], we are introducing the use of RandNLA methods into BLAS and LAPACK, leading to RandBLAS and RandLAPACK [69]. RandBLAS will be a library that concerns basic sketching for dense data matrices. RandLAPACK will be a library that concerns algorithms for solving traditional linear algebra problems and advanced sketching functionality.

The connection with and value of Modern RandNLA for RandBLAS/RandLAPACK (and other such software projects) is subtle but important: *due to the use of RMT-based ideas, in particular Def. 2 of a nearly-unbiased estimator, we can obtain a much smaller theory-practice gap than was possible with Classical RandNLA, which depended on the notion of a subspace embedding from Def. 1, for a broad range of implementations* [16, 52, 53]. For implementations in the past, one often used expensive Gaussian random projections to obtain stronger theory; and then implementations may or may not have used Gaussian random projections; thus leading to a potentially-large theory practice gap. However, with Modern RandNLA, e.g., having stronger control on the inversion bias (5), one can hope to analyze the distribution being implemented.

Developing software to go beyond BLAS/LAPACK to RandBLAS/RandLAPACK, as well as beyond single-machine shared-memory is a large topic. This is described in more details in the full technical report [22]; see also the monograph [69].

7 CONCLUDING THOUGHTS

There are many topics we did not cover: data-driven methods for learning good sketching operators; RandNLA methods for tensor decompositions; RandNLA methods in streaming/online environments; probabilistic numerics; as well as topics such as Hutchinson and spectral function methods, using randomness deep inside an algorithm, e.g., for pivot rule decisions, using these methods for the theory/practice of neural networks, e.g., NTK and Nystromformer, using these methods for quantization and low-precision computation, etc. In many of these cases, it is increasingly-important to identify core linear algebraic structures and build algorithmic/statistical methods around them (rather than "tacking" them on later as a "band aid" to fix problems that arise when one ignores key linear algebraic issues). This will be increasingly-important as heterogeneous hardware trends and ML training trends continue to grow. We expect the "Modern" theoretical tools developed by/for RandNLA that we have covered will be useful in these and other situations.

REFERENCES

[1] Naman Agarwal, Brian Bullins, and Elad Hazan. 2017. Second-order stochastic optimization for machine learning in linear time. *The Journal of Machine Learning Research* 18, 1 (2017), 4148–4187.

[2] Nir Ailon and Bernard Chazelle. 2009. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on computing* 39, 1 (2009), 302–322.

[3] Haim Avron, Petar Maymounkov, and Sivan Toledo. 2010. Blendenpik: Supercharging LAPACK's Least-Squares Solver. *SIAM Journal on Scientific Computing* 32, 3 (2010), 1217–1236.

[4] Haim Avron, Esmond Ng, and Sivan Toledo. 2009. Using Perturbed QR Factorizations to Solve Linear Least-Squares Problems. *SIAM J. Matrix Anal. Appl.* 31, 2 (2009), 674–693.

[5] J. Baik and J. W. Silverstein. 2006. Eigenvalues of large sample covariance matrices of spiked population models. *Journal of Multivariate Analysis* 97, 6 (2006), 1382–1408.

[6] Albert S Berahas, Raghu Bollapragada, and Jorge Nocedal. 2020. An investigation of Newton-sketch and subsampled Newton methods. *Optimization Methods and Software* 35, 4 (2020), 661–680.

[7] Raghu Bollapragada, Richard H Byrd, and Jorge Nocedal. 2018. Exact and inexact subsampled Newton methods for optimization. *IMA J. Numer. Anal.* 39, 2 (2018).

[8] Leon Bottou and Olivier Bousquet. 2007. The Tradeoffs of Large Scale Learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'07)*. Curran Associates Inc., USA, 161–168.

[9] Tatiana Brailovskaya and Ramon van Handel. 2022. Universality and sharp matrix concentration inequalities. *arXiv preprint arXiv:2201.05142* (2022).

[10] Shabarish Chenakkod, Michał Dereziński, Xiaoyu Dong, and Mark Rudelson. 2024. Optimal Embedding Dimension for Sparse Subspace Embeddings. *Symposium on Theory of Computing (STOC)* (2024).

[11] Kenneth L. Clarkson and David P. Woodruff. 2017. Low-Rank Approximation and Regression in Input Sparsity Time. *J. ACM* 63, 6, Article 54 (Jan. 2017), 45 pages.

[12] Michael B Cohen, Jelani Nelson, and David P Woodruff. 2016. Optimal Approximate Matrix Product in Terms of Stable Rank. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik.

[13] J. Demmel, J. Dongarra, J. Langou, J. Langou, P. Luszczek, and M. W. Mahoney. 2020. *Prospectus for the Next LAPACK and ScaLAPACK Libraries: Basic ALgebra LIbraries for Sustainable Technology with Interdisciplinary Collaboration (BALLISTIC)*. Technical Report LAWNs (LAPACK Working Notes), lawn297, ICL-UT-20-07.

[14] Michał Dereziński. 2022. Stochastic Variance-Reduced Newton: Accelerating Finite-Sum Minimization with Large Batches. *arXiv preprint arXiv:2206.02702* (2022).

[15] Michał Dereziński. 2023. Algorithmic gaussianization through sketching: Converting data into sub-Gaussian random designs. In *The Thirty Sixth Annual Conference on Learning Theory*. PMLR, 3137–3172.

[16] Michał Dereziński, Burak Bartan, Mert Pilanci, and Michael W Mahoney. 2020. Debiasing distributed second order optimization with surrogate sketching and scaled regularization. *Advances in Neural Information Processing Systems* 33 (2020), 6684–6695.

[17] Michał Dereziński, Jonathan Lacotte, Mert Pilanci, and Michael W Mahoney. 2021. Newton-LESS: Sparsification without Trade-offs for the Sketched Newton Update. *Advances in Neural Information Processing Systems* 34 (2021), 2835–2847.

[18] Michał Dereziński, Daniel LeJeune, Deanna Needell, and Elizaveta Rebrova. 2024. Fine-grained Analysis and Faster Algorithms for Iteratively Solving Linear Systems. *arXiv preprint arXiv:2405.05818* (2024).

[19] Michał Dereziński, Feynman T Liang, Zhenyu Liao, and Michael W Mahoney. 2020. Precise expressions for random projections: Low-rank approximation and randomized Newton. *Advances in Neural Information Processing Systems* 33 (2020).

[20] Michał Dereziński, Zhenyu Liao, Edgar Dobriban, and Michael Mahoney. 2021. Sparse sketches with small inversion bias. In *Conference on Learning Theory*. PMLR, 1467–1510.

[21] Michał Dereziński and Michael W Mahoney. 2021. Determinantal Point Processes in Randomized Numerical Linear Algebra. *Notices of the American Mathematical Society* 68, 1 (2021), 34–45.

[22] Michał Dereziński and Michael W. Mahoney. 2024. *Recent and Upcoming Developments in Randomized Numerical Linear Algebra for Machine Learning*. Technical Report arXiv:2406.11151. https://arxiv.org/pdf/2406.11151

[23] Michał Dereziński, Christopher Musco, and Jiaming Yang. 2024. Faster Linear Systems and Matrix Norm Approximation via Multi-level Sketched Preconditioning. *arXiv preprint arXiv:2405.05865* (2024).

[24] Michał Dereziński and Elizaveta Rebrova. 2024. Sharp analysis of sketch-and-project methods via a connection to randomized singular value decomposition. *SIAM Journal on Mathematics of Data Science* 6, 1 (2024), 127–153.

[25] Michał Dereziński and Jiaming Yang. 2024. Solving linear systems faster than via preconditioning. In *Proceedings of the Symposium on Theory of Computing (STOC)*.

[26] Edgar Dobriban and Sifan Liu. 2019. Asymptotics for sketching in least squares regression. *Advances in Neural Information Processing Systems* 32 (2019).

[27] P. Drineas, R. Kannan, and M. W. Mahoney. 2006. Fast Monte Carlo Algorithms for Matrices I: Approximating Matrix Multiplication. *SIAM J. Comput.* 36 (2006), 132–157.

[28] Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. 2012. Fast Approximation of Matrix Coherence and Statistical Leverage. *J. Mach. Learn. Res.* 13, 1 (2012), 3475–3506.

[29] P. Drineas and M. W. Mahoney. 2016. RandNLA: Randomized Numerical Linear Algebra. *Commun. ACM* 59 (2016), 80–90.

[30] P. Drineas and M. W. Mahoney. 2018. Lectures on Randomized Numerical Linear Algebra. In *The Mathematics of Data*. AMS/IAS/SIAM, 1–48.

[31] Petros Drineas, Michael W Mahoney, and S Muthukrishnan. 2006. Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. 1127–1136.

[32] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. 2008. Relative-Error CUR Matrix Decompositions. *SIAM J. Matrix Anal. Appl.* 30, 2 (2008), 844–881.

[33] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós. 2010. Faster Least Squares Approximation. *Numer. Math.* 117, 2 (2010), 219–249.

[34] David Durfee, Kevin A Lai, and Saurabh Sawlani. 2018. L1 Regression using Lewis Weights Preconditioning and Stochastic Gradient Descent. In *Conference On Learning Theory*. PMLR, 1626–1656.

[35] Murat A Erdogdu and Andrea Montanari. 2015. Convergence rates of sub-sampled Newton methods. *Advances in Neural Information Processing Systems* 28 (2015), 3052–3060.

[36] Zachary Frangella, Pratik Rathore, Shipu Zhao, and Madeleine Udell. 2023. PROMISE: Preconditioned Stochastic Optimization Methods by Incorporating Scalable Curvature Estimates. *arXiv preprint arXiv:2309.02014* (2023).

[37] Sachin Garg, Kevin Tan, and Michał Dereziński. 2024. Distributed Least Squares in Small Space via Sketching and Bias Reduction. *arXiv preprint arXiv:2405.05343* (2024).

[38] A. Gittens, J. Kottalam, J. Yang, M. F. Ringenburg, J. Chhugani, E. Racah, M. Singh, Y. Yao, C. Fischer, O. Ruebel, B. Bowen, N. G. Lewis, M. W. Mahoney, V. Krishnamurthy, and Prabhat. 2016. A multi-platform evaluation of the randomized CX low-rank matrix factorization in Spark. In *Proc. 5th International Workshop on Parallel and Distributed Computing for Large Scale Machine Learning and Big Data Analytics, at IPDPS*.

[39] Alon Gonen, Francesco Orabona, and Shai Shalev-Shwartz. 2016. Solving ridge regression using sketched preconditioned SVRG. In *International conference on machine learning*. PMLR, 1397–1405.

[40] Robert Gower, Filip Hanzely, Peter Richtárik, and Sebastian U Stich. 2018. Accelerated stochastic matrix inversion: general theory and speeding up BFGS rules for faster second-order optimization. *Advances in Neural Information Processing Systems* 31 (2018).

[41] Robert Gower, Dmitry Kovalev, Felix Lieder, and Peter Richtárik. 2019. RSN: Randomized subspace newton. *Advances in Neural Information Processing Systems* 32 (2019).

[42] Robert Gower, Nicolas Le Roux, and Francis Bach. 2018. Tracking the gradients using the Hessian: A new look at variance reducing stochastic methods. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 707–715.

[43] Robert M Gower and Peter Richtárik. 2015. Randomized iterative methods for linear systems. *SIAM J. Matrix Anal. Appl.* 36, 4 (2015), 1660–1690.

[44] Vipul Gupta, Avishek Ghosh, Michał Dereziński, Rajiv Khanna, Kannan Ramchandran, and Michael W Mahoney. 2021. LocalNewton: Reducing communication rounds for distributed learning. In *Uncertainty in Artificial Intelligence*. PMLR, 632–642.

[45] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review* 53, 2 (2011), 217–288.

[46] Filip Hanzely, Nikita Doikov, Yurii Nesterov, and Peter Richtarik. 2020. Stochastic subspace cubic Newton method. In *International Conference on Machine Learning*. PMLR, 4027–4038.

[47] Filip Hanzely, Konstantin Mishchenko, and Peter Richtárik. 2018. SEGA: Variance reduction via gradient sketching. *Advances in Neural Information Processing Systems* 31 (2018).

[48] L. Hodgkinson and M. W. Mahoney. 2020. *Multiplicative noise and heavy tails in stochastic optimization*. Technical Report Preprint: arXiv:2006.06293.

[49] M. S. Kaczmarz. 1937. Angenaherte Auflosung von Systemen linearer Gleichungen. *Bulletin International de l'Academie Polonaise des Sciences et des Lettres* 35 (1937), 355–357.

[50] R. Kannan and S. Vempala. 2017. Randomized algorithms in numerical linear algebra. *Acta Mathematica* 26 (2017), 95–135.

[51] Dmitry Kovalev, Konstantin Mishchenko, and Peter Richtárik. 2019. Stochastic Newton and cubic Newton methods with simple local linear-quadratic rates. *arXiv preprint arXiv:1912.01597* (2019).

[52] Jonathan Lacotte, Sifan Liu, Edgar Dobriban, and Mert Pilanci. 2020. Optimal iterative sketching methods with the subsampled randomized Hadamard transform. *Advances in Neural Information Processing Systems* 33 (2020), 9725–9735.

[53] Jonathan Lacotte and Mert Pilanci. 2022. Adaptive and oblivious randomized subspace methods for high-dimensional optimization: Sharp analysis and lower bounds. *IEEE Transactions on Information Theory* 68, 5 (2022), 3281–3303.

[54] Yin Tat Lee and Aaron Sidford. 2013. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *2013 ieee 54th annual symposium on foundations of computer science*. IEEE, 147–156.

[55] Dennis Leventhal and Adrian S Lewis. 2010. Randomized methods for linear constraints: convergence rates and conditioning. *Mathematics of Operations Research* 35, 3 (2010), 641–654.

[56] Yanli Liu, Fei Feng, and Wotao Yin. 2019. Acceleration of SVRG and Katyusha X by Inexact Preconditioning. In *International Conference on Machine Learning*. PMLR, 4003–4012.

[57] Ping Ma, Yongkai Chen, Xinlian Zhang, Xin Xing, Jingyi Ma, and Michael W Mahoney. 2022. Asymptotic analysis of sampling estimators for randomized numerical linear algebra algorithms. *The Journal of Machine Learning Research* 23, 1 (2022), 7970–8014.

[58] P. Ma, M. W. Mahoney, and B. Yu. 2015. A Statistical Perspective on Algorithmic Leveraging. *Journal of Machine Learning Research* 16 (2015), 861–911.

[59] M. W. Mahoney. 2011. *Randomized algorithms for matrices and data*. NOW Publishers, Boston.

[60] M. W. Mahoney. 2016. *Lecture Notes on Randomized Linear Algebra*. Technical Report Preprint: arXiv:1608.04481.

[61] C. H. Martin and M. W. Mahoney. 2018. *Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning*. Technical Report Preprint: arXiv:1810.01075.

[62] P.-G. Martinsson. 2018. Randomized methods for matrix computations. In *The Mathematics of Data*, M. W. Mahoney, J. C. Duchi, and A. C. Gilbert (Eds.). AMS/IAS/SIAM, 187–230.

[63] Per-Gunnar Martinsson and Joel A Tropp. 2020. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica* 29 (2020), 403–572.

[64] J. Matousek. 2008. On variants of the Johnson–Lindenstrauss lemma. *Random Structures and Algorithms* 33, 2 (2008), 142–156.

[65] Xiangrui Meng and Michael W. Mahoney. 2013. Low-distortion Subspace Embeddings in Input-sparsity Time and Applications to Robust Linear Regression. In *Proceedings of the Symposium on Theory of Computing (STOC '13)*. 91–100.

[66] X. Meng, M. A. Saunders, and M. W. Mahoney. 2014. LSRN: A Parallel Iterative Solver for Strongly Over- or Under-Determined Systems. *SIAM Journal on Scientific Computing* 36, 2 (2014), C95–C118.

[67] Raphael A Meyer, Cameron Musco, Christopher Musco, and David P Woodruff. 2021. Hutch++: Optimal stochastic trace estimation. In *Symposium on Simplicity in Algorithms (SOSA)*. SIAM, 142–155.

[68] Aryan Mokhtari, Mark Eisen, and Alejandro Ribeiro. 2018. IQN: An incremental quasi-Newton method with local superlinear convergence rate. *SIAM Journal on Optimization* 28, 2 (2018), 1670–1698.

[69] R. Murray, J. Demmel, M. W. Mahoney, N. B. Erichson, M. Melnichenko, O. A. Malik, L. Grigori, P. Luszczek, M. Dereziński, M. E. Lopes, T. Liang, H. Luo, and J. Dongarra. 2023. *Randomized Numerical Linear Algebra – A Perspective on the Field with an Eye to Software*. Technical Report Preprint: arXiv:2302.11474v2.

[70] Mojmir Mutny, Michał Dereziński, and Andreas Krause. 2020. Convergence Analysis of Block Coordinate Algorithms with Determinantal Sampling. In *International Conference on Artificial Intelligence and Statistics*. 3110–3120.

[71] Sen Na, Michał Dereziński, and Michael W Mahoney. 2022. Hessian averaging in stochastic Newton methods achieves superlinear convergence. *Mathematical Programming* (2022), 1–48.

[72] Deanna Needell and Joel A Tropp. 2014. Paved with good intentions: analysis of a randomized block Kaczmarz method. *Linear Algebra Appl.* 441 (2014), 199–221.

[73] Deanna Needell and Rachel Ward. 2013. Two-subspace projection method for coherent overdetermined systems. *Journal of Fourier Analysis and Applications* 19, 2 (2013), 256–269.

[74] Deanna Needell, Rachel Ward, and Nati Srebro. 2014. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Advances in neural information processing systems* 27 (2014).

[75] Jelani Nelson and Huy L. Nguyên. 2013. OSNAP: Faster Numerical Linear Algebra Algorithms via Sparser Subspace Embeddings. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS '13)*. 117–126.

[76] M. Pilanci and M. J. Wainwright. 2016. Iterative Hessian Sketch: Fast and Accurate Solution Approximation for Constrained Least-Squares. *Journal of Machine Learning Research* 17, 53 (2016), 1–38.

[77] Mert Pilanci and Martin J Wainwright. 2017. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization* 27, 1 (2017), 205–245.

[78] G. Raskutti and M. W. Mahoney. 2016. A Statistical Perspective on Randomized Sketching for Ordinary Least-Squares. *Journal of Machine Learning Research* 17, 214 (2016), 1–31.

[79] Elizaveta Rebrova and Deanna Needell. 2021. On block Gaussian sketching for the Kaczmarz method. *Numerical Algorithms* 86 (2021), 443–473.

[80] Anton Rodomanov and Dmitry Kropotov. 2020. A randomized coordinate descent method with volume sampling. *SIAM Journal on Optimization* 30, 3 (2020), 1878–1904.

[81] Vladimir Rokhlin and Mark Tygert. 2008. A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences* 105, 36 (2008), 13212–13217.

[82] Farbod Roosta-Khorasani and Michael W Mahoney. 2019. Sub-sampled Newton methods. *Mathematical Programming* 174 (2019), 293–326.

[83] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. 2020. Fetchsgd: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*. PMLR, 8253–8265.

[84] Mark Rudelson and Roman Vershynin. 2013. Hanson-Wright inequality and sub-gaussian concentration. *Electronic Communications in Probability* 18 (2013).

[85] Tamas Sarlos. 2006. Improved Approximation Algorithms for Large Matrices via Random Projections. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS '06)*. 143–152.

[86] G.W. Stewart. 1977. On the Perturbation of Pseudo-Inverses, Projections and Linear Least Squares Problems. *SIAM Rev.* 19 (1977), 634–662.

[87] Thomas Strohmer and Roman Vershynin. 2009. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications* 15, 2 (2009), 262–278.

[88] J. A. Tropp. 2015. *An introduction to matrix concentration inequalities*. NOW Publishers, Boston.

[89] David P Woodruff. 2014. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science* 10, 1–2 (2014), 1–157.

[90] P. Xu, J. Yang, F. Roosta-Khorasani, C. Re, and M. W. Mahoney. 2016. *Sub-sampled Newton Methods with Non-uniform Sampling*. Technical Report Preprint: arXiv:1607.00559.

[91] Jiyan Yang, Yin-Lam Chow, Christopher Ré, and Michael W Mahoney. 2017. Weighted SGD for lp regression with randomized preconditioning. *The Journal of Machine Learning Research* 18, 1 (2017), 7811–7853.

[92] J. Yang, X. Meng, and M. W. Mahoney. 2016. Implementing Randomized Matrix Algorithms in Parallel and Distributed Environments. *Proc. IEEE* 104, 1 (2016), 58–92.

[93] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. 2020. PyHessian: Neural networks through the lens of the Hessian. In *2020 IEEE international conference on big data (Big data)*. IEEE, 581–590.

[94] Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael W Mahoney. 2021. AdaHessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, Vol. 35.

[95] Rui Yuan, Alessandro Lazaric, and Robert M Gower. 2022. Sketched Newton–Raphson. *SIAM Journal on Optimization* 32, 3 (2022), 1555–1583.