Journal of Machine Learning Research 26 (2025) 1-49 Submitted 11/24; Revised 4/25; Published 6/25

Fine-grained Analysis and Faster Algorithms for Iteratively Solving Linear Systems

Michał Dereziński DEREZIN@UMICH.EDU
2260 Hayward St., Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI, 48109 USA

Daniel LeJeune DANIEL@DLEJ.NET
390 Jane Stanford Way, Department of Statistics
Stanford University, Palo Alto, CA, 94305 USA

Deanna Needell DEANNA@MATH.UCLA.EDU
520 Portola Plaza, Department of Mathematics
University of California, Los Angeles, CA, 90095 USA

Elizaveta Rebrova ELRE@PRINCETON.EDU
98 Charlton St., Department of Operations Research and Financial Engineering
Princeton University, Princeton, NJ, 08540 USA

Editor: Zhihua Zhang

Abstract

Despite being a key bottleneck in many machine learning tasks, the cost of solving large linear systems has proven challenging to quantify due to problem-dependent quantities such as condition numbers. To tackle this, we consider a fine-grained notion of complexity for solving linear systems, which is motivated by applications where the data exhibits low-dimensional structure, including spiked covariance models and kernel machines, and when the linear system is explicitly regularized, such as ridge regression.

Concretely, let κ_ℓ be the ratio between the ℓth largest and the smallest singular value of $n\times n$ matrix $\mathbf{A}$. We give a stochastic algorithm based on the Sketch-and-Project paradigm, that solves the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ in time $\tilde{O}(\kappa_\ell \cdot n^2 \log 1/\epsilon)$ for any $\ell = O(n^{0.729})$. This is a direct improvement over preconditioned conjugate gradient, and it provides a stronger separation between stochastic linear solvers and algorithms accessing $\mathbf{A}$ only through matrix-vector products.

Our main technical contribution is the new analysis of the first and second moments of the random projection matrix that arises in Sketch-and-Project.

Keywords: Linear systems, matrix sketching, sketch-and-project, stochastic optimization, random matrix theory

1. Introduction

In the era of big data, the efficient processing of massive data sets has become critically important across a wide range of areas, from machine learning and statistics to scientific computing and industrial applications. Traditional methods, and especially direct approaches, for handling such data often face significant computational challenges due to their high dimensionality and massive volume. In response to this, iterative refinement methods using randomized sampling and sketching have emerged as powerful tools for effectively solving algorithmic tasks in large-scale machine learning and data science. Yet, the computational cost of these methods is often significantly affected by problem-dependent quantities such as condition numbers, which make it challenging to characterize how their complexity compares to, and is affected by, recent advances in algorithmic theory.

Perhaps one of the most fundamental tasks impacted by this phenomenon is solving large systems of linear equations, which has numerous applications in machine learning such as least squares (Dieuleveut et al., 2017), kernel ridge regression (Alaoui and Mahoney, 2015), as well as model training with Newton-type methods on both convex and non-convex objectives (Erdogdu and Montanari, 2015; Xu et al., 2020). Other applications include imaging (Natterer, 2001; Hounsfield, 1973), sensor networks (Savvides et al., 2001), and scientific computing (Xia et al., 2010; Wolters et al., 2008), among others. In this problem, our goal is to approximately solve $\mathbf{A}\mathbf{x} = \mathbf{b}$, given a large data matrix $\mathbf{A}$ and a vector $\mathbf{b}$. Traditional direct approaches for solving linear systems, such as Gaussian elimination, require $O(n^3)$ time to find the exact solution when $\mathbf{A}$ is a dense square $n \times n$ matrix. Compared to this, deterministic iterative refinement methods, such as Richardson or Chebyshev iteration (Golub and Varga, 1961) and Krylov methods including the Lanczos algorithm and Conjugate Gradient (Saad, 1981; Liesen and Strakos, 2013), produce a sequence of estimates which gradually converge to the solution, having a much cheaper $O(n^2)$ per-iteration cost that comes typically from computing a matrix-vector product with $\mathbf{A}$. Introducing sub-sampling into the iterations has led to stochastic approaches such as Randomized Kaczmarz (Kaczmarz, 1937; Strohmer and Vershynin, 2009) and Randomized Coordinate Descent (Nesterov, 2012), which have an even smaller per-iteration cost but tend to require more steps to converge.

Another algorithmic approach, which has led to improvements in the time complexity of solving linear systems, is fast matrix multiplication. Initiated by Strassen (1969), this approach relies on the fact that we can invert an $n \times n$ matrix in the time that it takes to multiply two such matrices. This has led to algorithms with runtime of $O(n^\omega)$, where $\omega < 2.371552$ is the current exponent of matrix multiplication, which is regularly improved with continued advances in the area (Pan, 1984; Coppersmith and Winograd, 1987; Williams, 2012; Williams et al., 2024). Unfortunately, except in special cases (Peng and Vempala, 2021), these algorithmic advances have not led to improvements in the complexity of iterative linear system solvers, due to the fundamentally sequential nature of these methods, as well as their dependence on the condition number κ.

The traditional analysis of iterative methods runs into a fundamental complexity barrier of $\tilde{O}(\kappa \cdot n^2)$, or $\tilde{O}(\sqrt{\kappa} \cdot n^2)$ in the positive definite setting, which depends on the condition number κ of the matrix $\mathbf{A}$, defined as the ratio between its largest and smallest singular value: $\kappa = \sigma_1/\sigma_n$, where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$ are the decreasing singular values of $\mathbf{A}$. Yet, a single condition number does not accurately characterize the cost of iteratively solving a linear system, particularly in machine learning settings where the singular value profile of the input matrix exhibits low-dimensional structure determined by the underlying data distribution or a kernel function.

In this paper, we study the time complexity of iterative linear system solvers through a more fine-grained notion than a single condition number quantity, in a way that is particularly well-suited for machine learning and statistical settings. Concretely, we consider a *parameterized* condition number κ_ℓ, for $\ell \in \{1, ..., n\}$, which allows the top-ℓ part of the spectrum to be arbitrarily ill-conditioned, while controlling the condition number of the tail of the spectrum (related notions of condition number have been considered, see Section 2). In this model, we obtain improved time complexity guarantees for solving linear systems (see Theorems 1 and 2) through a combination of new convergence analysis and efficient algorithms for stochastic iterative solvers based on the Sketch-and-Project paradigm (Gower and Richtárik, 2015a; Gower et al., 2018). Our approach not only provides sharper guarantees for wide classes of matrices, but also enables us to tie the complexity of iterative solvers together with the ongoing algorithmic advances in fast matrix multiplication, as well as providing a stronger complexity separation between stochastic iterative methods and classical iterative algorithms such as conjugate gradient (see Theorem 3).

Our fine-grained analysis is well-motivated by many established statistical models of data matrices. These models are commonly of the form "signal+noise", which corresponds to the intuition that most of the information is contained in a low-dimensional component of the data. A classical example of this is the spiked covariance model (Johnstone, 2001; Capitaine et al., 2009; Cai et al., 2013; Perry et al., 2018), which describes the data as a low-rank matrix distorted by noise (i.e., $\mathbf{A} + \epsilon\mathbf{G}$). Moreover, linear systems are often regularized before solving (i.e., $\mathbf{A} + \lambda\mathbf{I}$), either to achieve better generalization, e.g., for kernel ridge regression (Alaoui and Mahoney, 2015), or to attain improved convergence in an optimization method, e.g., damped or cubic-regularized Newton's method (Nesterov and Polyak, 2006).

Other motivating examples include matrices arising from feature extraction techniques commonly used in machine learning, such as kernel machines (Williams and Seeger, 2001), Gaussian processes (Rasmussen and Williams, 2006), and random features (Rahimi and Recht, 2007), which lead to well-understood polynomial or exponential singular value decay profiles (Burt et al., 2019; Santa et al., 1997; Rasmussen and Williams, 2006). For example, our new results imply improved runtimes for solving linear systems with a polynomial spectral decay $\sigma_i \simeq i^{-\beta}$ (see Corollary 4), which arise when using certain kernel functions (e.g., Matérn, Rasmussen and Williams, 2006), and also, as a result of the power law.

1.1 Main Results

Our main result shows that the $\tilde{O}(\kappa \cdot n^2)$ time complexity of iteratively solving a linear system can be directly improved by replacing the condition number $\kappa = \sigma_1/\sigma_n$ with the *spectral tail condition number*, $\kappa_\ell := \sigma_\ell/\sigma_n$ where σ_ℓ is the ℓth largest singular value of $\mathbf{A}$. Using fast matrix multiplication, we show that this can be achieved with any $\ell = O(n^{0.729})$.

Theorem 1 *Consider an $n \times n$ matrix $\mathbf{A}$ with singular values $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n > 0$ and an n-dimensional vector $\mathbf{b}$. We can compute $\tilde{\mathbf{x}}$ such that $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\| \leq \epsilon\|\mathbf{b}\|$ in time:*

$$\tilde{O}\bigg(\frac{\sigma_\ell}{\sigma_n} \cdot n^2 \log 1/\epsilon\bigg) \quad \textit{for any} \quad \ell = O\big(n^{\frac{1}{\omega-1}}\big) = O(n^{0.729}).$$

Note that the complexity improvement in this result is already significant without resorting to fast matrix multiplication, in which case, we can use any $\ell = O(\sqrt{n})$. Moreover, unlike with the usual condition number κ, any future improvements in the matrix multiplication exponent ω will directly lead to improvements in the exponent of ℓ in κ_ℓ. As ω approaches 2, similarly ℓ approaches n, until the two paradigms (potentially) meet at $\tilde{O}(n^2)$.

To underline that this fine-grained notion of complexity is natural, we show that the above guarantee holds for a remarkably simple stochastic iterative solver (see Algorithm 1), a combination of two standard techniques: Sketch-and-Project (Gower and Richtárik, 2015a), which is a randomized iterative framework for solving linear systems that can be viewed as a simple extension of the classical Kaczmarz algorithm (Kaczmarz, 1937; Strohmer and Vershynin, 2009); and Nesterov's acceleration technique (Nesterov, 1983), variants of which have been used in numerous iterative optimization methods (Liu and Wright, 2016; Ye et al., 2020; Even et al., 2021). While this combination is not new (Gower et al., 2018), its convergence has proven challenging to quantify in terms of the spectral properties of $\mathbf{A}$.

We address this challenge by bridging ultra-sparse sketching techniques (Chenakkod et al., 2024) with a new convergence analysis that builds on recent advances in combinatorial sampling (Dereziński et al., 2020) and random matrix theory (Brailovskaya and van Handel, 2024). Along the way, we establish technical results that are likely of independent interest, including a new characterization of the smallest singular value for the family of random matrices obtained via sparse sketching (Lemma 22).

Positive definite systems. Our algorithmic framework can be adapted to take advantage of certain types of additional structure present in the matrix $\mathbf{A}$. As an example, we show that, for positive definite matrices, the dependence on the spectral tail condition number κ_ℓ can be improved to a square root, just as regular condition number κ can be improved to a square root for deterministic iterative methods (Axelsson and Lindskog, 1986).

Theorem 2 *Consider an $n \times n$ positive definite $\mathbf{A}$ with singular values $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$ and an n-dimensional vector $\mathbf{b}$. We can compute $\tilde{\mathbf{x}}$ such that $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\| \leq \epsilon\|\mathbf{b}\|$ in time:*

$$\tilde{O}\left(\sqrt{\frac{\sigma_\ell}{\sigma_n}} \cdot n^2 \log 1/\epsilon\right) \quad \textit{for any} \quad \ell = O\big(n^{\frac{1}{\omega-1}}\big) = O(n^{0.729}).$$

Separation from matrix-vector query methods. An important implication of our results, and of our reliance on the spectral tail condition number, is a more distinct than previously known separation (in terms of complexity) between stochastic solvers, such as Sketch-and-Project, and classical iterative solvers, which interact with $\mathbf{A}$ only through matrix-vector product queries. Here, representing state-of-the-art matrix-vector query solvers, let us consider Krylov methods, such as Conjugate Gradient (CG, Hestenes et al., 1952) and its non-symmetric extensions (Saad, 1981). A careful convergence analysis of Krylov iterations shows that, given any ℓ and assuming exact precision arithmetic, they converge ϵ-close in $O(n^2\ell + \kappa_\ell \cdot n^2 \log 1/\epsilon)$ time for general linear systems, and in $O(n^2\ell + \sqrt{\kappa_\ell} \cdot n^2 \log 1/\epsilon)$ time for positive definite systems (Axelsson and Lindskog, 1986; Spielman and Woo, 2009). In both cases, the second term matches our complexity in Theorems 1 and 2, but the Krylov methods suffer an additional cost of $O(n^2\ell)$ to take care of the top ℓ singular values of $\mathbf{A}$.

In fact, we show that this additional cost is a fundamental bottleneck of all methods based on the matrix-vector query model. Specifically, we give a lower bound showing that any algorithm which accesses the matrix $\mathbf{A}$ only through matrix-vector product queries must incur a worst-case cost of $\tilde{\Omega}(n^2\ell + \sqrt{\kappa_\ell} \cdot n^2)$ arithmetic operations for dense positive definite linear systems. See Section 9 for a detailed discussion.

Theorem 3 *Given n, $\ell < n$, and $\kappa_\ell \geq 1$, consider the task of solving an $n \times n$ positive definite linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\frac{\sigma_\ell(\mathbf{A})}{\sigma_n(\mathbf{A})} \leq \kappa_\ell$. Any algorithm which interacts with $\mathbf{A}$ via adaptive randomized queries of the form $\mathbf{v} \to \mathbf{A}\mathbf{v}$, and solves this task to high precision, has query complexity at least $\tilde{\Omega}(\ell + \sqrt{\kappa_\ell})$, which for dense systems leads to time complexity at least $\tilde{\Omega}(n^2\ell + \sqrt{\kappa_\ell} \cdot n^2)$.*

Note that, in addition to deterministic solvers like CG, this lower bound also includes randomized preconditioning methods (Martinsson and Tropp, 2020), which probe matrix $\mathbf{A}$ with Gaussian vectors to construct a low-rank approximation (essentially, estimating the top part of its spectrum), and use that information to speed up an iterative solver like CG. Once we go beyond the matrix-vector query model, this preconditioning approach can be sped up by blocking the queries together and using fast $n \times n$ by $n \times \tilde{O}(\ell)$ rectangular matrix multiplication (Le Gall, 2012) to approximate the top part of the spectrum faster. However, even then, the cost is still $\Omega(n^{2+\theta})$ with some $\theta > 0$ for any $\ell = \Omega(n^{0.33})$. In comparison, Theorems 1 and 2 show that Sketch-and-Project with Nesterov's acceleration avoids this cost entirely for any $\ell = O(n^{0.729})$.

Application: Polynomial spectral decay. To illustrate how our new results improve on the time complexity of solving linear systems for real-world matrices, we consider matrices with polynomial spectral decay (power law distribution). Concretely, consider a matrix $\mathbf{A}$ with singular values $\sigma_i = \Theta(i^{-\beta}\sigma_1)$ for some positive constant β. Such behavior can naturally occur in data (Clauset et al., 2009; Eikmeier and Gleich, 2017), or it can arise when $\mathbf{A}$ is the kernel matrix of a data set, i.e., its (i,j)th entry is $k(x_i, x_j)$ where x_i is the ith data point and $k(\cdot,\cdot)$ is a kernel function. For example, Rasmussen and Williams (2006) showed that if we use the Matérn kernel function with parameter $\nu > 0$, then matrix $\mathbf{A}$ exhibits polynomial decay with $\beta = 2\nu+1$. Solving linear systems with these matrices is the main computational cost in kernel ridge regression, Gaussian processes, and independent component analysis, among others. Our results imply the following guarantee for solving linear systems with polynomial decay.

Corollary 4 *Consider an $n \times n$ matrix $\mathbf{A}$ with singular values $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$ and an n-dimensional vector $\mathbf{b}$. Suppose that $c_1 i^{-\beta} \leq \sigma_i \leq c_2 i^{-\beta}$ for some absolute constants $0 < c_1 < c_2$ and $\beta \geq 0.5$. Then, we can compute $\tilde{\mathbf{x}}$ such that $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\| \leq \epsilon\|\mathbf{b}\|$ in time:*

$$\tilde{O}\Big(n^{2+\frac{\omega-2}{\omega-1}(\beta-0.5)} \log 1/\epsilon\Big).$$

Remark 1 *This is better than the best known guarantee in the matrix-vector query model (Spielman and Woo, 2009) for any $\beta \in (0.5, 3.75)$, and it improves on the prior best known time complexity in the unrestricted model for any $\beta \in (0.5, 1.33)$. For example, if we choose $\beta = 1$, then we get $\tilde{O}(n^{2.135})$ runtime, whereas the best prior result (obtained by preconditioning an SVRG-type stochastic solver, see Gonen et al., 2016; Musco et al., 2018b) achieves $\tilde{O}(n^{2.178})$, and CG obtains $\tilde{O}(n^{2.667})$. For a detailed discussion see Appendix B.*

Extensions to sparse least squares. While our algorithms are stated for consistent linear systems, they can be easily extended to the general inconsistent setting, i.e., least squares regression. This can be achieved by incorporating a variant of our Sketch-and-Project algorithm as an inner solver within a sketch-to-precondition type algorithm (Rokhlin and Tygert, 2008), obtaining nearly-input-sparsity time algorithms for a tall least squares task parameterized by the spectral tail condition number κ_ℓ. For example, given an $n \times d$ matrix $\mathbf{A}$ and an n-dimensional vector $\mathbf{b}$, we can find $\tilde{\mathbf{x}}$ such that $\|\mathbf{A}\tilde{\mathbf{x}}-\mathbf{b}\|^2 \leq \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 + \epsilon\|\mathbf{b}\|^2$ in $\tilde{O}((\mathrm{nnz}(\mathbf{A}) + d^2\kappa_\ell)\log 1/\epsilon)$ time, where $\mathrm{nnz}(\mathbf{A})$ is the number of non-zeros in $\mathbf{A}$ and $\kappa_\ell = \sigma_\ell(\mathbf{A})/\sigma_d(\mathbf{A})$ for $\ell = O(d^{0.729})$. See Section 8 for further discussion.

1.2 Main Technical Contributions

Our algorithms are built on the Sketch-and-Project framework (Gower and Richtárik, 2015a), an extension of the Randomized Kaczmarz algorithm (Strohmer and Vershynin, 2009). The core idea of Sketch-and-Project is that in each iteration we construct a small randomized sketch of the linear system, and then project the current iterate onto the set of solutions of that sketch. Formally, given an $n \times n$ linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ and a current iterate $\mathbf{x}_t$, we generate a new random $k \times n$ sketching matrix $\mathbf{S} = \mathbf{S}(t)$, where $k \ll n$ is the sketch size, and then consider a smaller linear system $\mathbf{S}\mathbf{A}\mathbf{x} = \mathbf{S}\mathbf{b}$. For example, $\mathbf{S}$ can be a subsampling matrix which selects k out of n linear equations (rows of $\mathbf{A}$). In the original Sketch-and-Project algorithm, we would now compute $\mathbf{x}_{t+1}$ as the projection of $\mathbf{x}_t$ onto the subspace defined by the smaller system. A natural extension of this is to use Nesterov's acceleration scheme (Gower et al., 2018), which introduces two additional sequences $\mathbf{y}_t$ and $\mathbf{v}_t$ to mix the projection step with the current trajectory of the iterates (see Algorithm 1).

While various Sketch-and-Project algorithms have been proposed and studied across a long line of works, such as Hanzely et al. (2018); Necoara et al. (2019); LeJeune et al. (2024); Tang et al. (2023), their complexity analysis has proven remarkably challenging. Crucially, their convergence rate depends on the spectral properties of a random projection matrix defined by the sketching matrix $\mathbf{S}$, namely $\mathbf{P} := (\mathbf{S}\mathbf{A})^\dagger\mathbf{S}\mathbf{A}$, which is a projection onto the subspace spanned by the rows of the sketch $\mathbf{S}\mathbf{A}$. Matrix $\mathbf{P}$ is also central to other applications of sketching, such as low-rank approximation (Halko et al., 2011; Cohen et al., 2015, 2017), and so understanding its spectral properties is of significant independent interest across the literature on randomized linear algebra (Woodruff, 2014; Martinsson and Tropp, 2020). Our main technical contributions are new sharp characterizations of the first and second matrix moments of $\mathbf{P}$ for a class of ultra-sparse sketching matrices $\mathbf{S}$, as explained below.

Sketch-and-Project with Nesterov's acceleration has been proposed in Richtárik and Takác (2020) and previously studied by Gower et al. (2018). However, these papers characterize the convergence rate of the method only in terms of the properties of the random projection $\mathbf{P}$, without identifying how these properties are determined by the spectrum of the input matrix $\mathbf{A}$, or the choice of the sketching matrix $\mathbf{S}$. Specifically, they showed that, as long as the matrix $\bar{\mathbf{P}} = \mathbb{E}[\mathbf{P}]$ is invertible, then the iterates converge to the solution $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$, with the expected convergence rate given by

$$\mathbb{E}\big[\|\mathbf{x}_t - \mathbf{x}^*\|^2\big] \leq 2\bigg(1 - \sqrt{\frac{\mu}{\nu}}\bigg)^t \|\mathbf{x}_0 - \mathbf{x}^*\|^2, \tag{1}$$
$$\text{where} \quad \mu = \lambda_{\min}(\mathbb{E}[\mathbf{P}]) \quad \text{and} \quad \nu = \lambda_{\max}\Big(\mathbb{E}\big[(\bar{\mathbf{P}}^{-1/2}\mathbf{P}\bar{\mathbf{P}}^{-1/2})^2\big]\Big).$$

Here, we can think of μ and ν as the first and second moment properties of the random projection matrix $\mathbf{P}$, since μ requires lower-bounding its first matrix moment $\bar{\mathbf{P}} = \mathbb{E}[\mathbf{P}]$, whereas ν requires upper-bounding its normalized second matrix moment $\mathbb{E}[(\bar{\mathbf{P}}^{-1/2}\mathbf{P}\bar{\mathbf{P}}^{-1/2})^2]$. The core challenge in obtaining these guarantees is that they are fundamentally average-case properties that cannot be recovered by usual high-dimensional concentration techniques (consider that we seek a bound on $\lambda_{\min}(\mathbb{E}[\mathbf{P}])$ when the matrix $\mathbf{P}$ is not even invertible).

While the characterization of the second-order term ν presented in this paper appears to be the first such guarantee, there has been a number of works analyzing the complexity of Sketch-and-Project (Rodomanov and Kropotov, 2020; Mutny et al., 2020; Dereziński and Rebrova, 2024; Dereziński and Yang, 2024) based on quantifying only the first-order term μ. These works do not consider Nesterov's acceleration and rely on a weaker convergence guarantee that follows from (1) by observing that $1-\sqrt{\mu/\nu}\leq 1-\mu$. This allows one to only have to lower bound μ, while avoiding the second matrix moment of $\mathbf{P}$, but it also leads to sub-optimal condition number dependence.

First such guarantees for μ were obtained only recently: initially for specialized combinatorial sub-sampling matrices $\mathbf{S}$ based on determinantal point processes (DPPs, see Rodomanov and Kropotov, 2020; Mutny et al., 2020); then, for dense Gaussian sketching matrices (Rebrova and Needell, 2021), as well as sub-Gaussian matrices with some degree of sparsity (Dereziński and Rebrova, 2024), and asymptotically free sketching matrices in the asymptotic regime (LeJeune et al., 2024). All of these sketching/sub-sampling approaches are still too expensive to yield fast algorithms, however the resulting guarantees exposed a connection between the first moment of $\mathbf{P}$ and the spectral tail condition number κ_ℓ when the sketch size k is proportional to ℓ.

Finally, Dereziński and Yang (2024) gave a reduction showing that after preprocessing the matrix $\mathbf{A}$ with a randomized Hadamard transform (Ailon and Chazelle, 2009), a uniform sub-sampling matrix $\mathbf{S}$ of size k yields a guarantee on μ that is at least as good as a DPP sub-sampling matrix of size $\ell=\Omega(k/\log^3 n)$. This led to the first *efficient* sub-sampling scheme with a convergence guarantee, although requiring the sample size to be larger by a factor of $O(\log^3 n)$ than what we might hope to achieve.

Technical contribution 1: *First-moment projection analysis via optimal DPP reduction.* In our first main technical contribution, we directly improve on the existing guarantees for the first matrix moment of the projection matrix $\mathbf{P}$, both for uniform sub-sampling matrices $\mathbf{S}$, as well as when using ultra-sparse sketching matrices. Our general strategy is similar to that of Dereziński and Yang (2024), in that we also show a reduction from uniform to DPP sub-sampling, however the actual reduction is entirely different (it involves designing a custom DPP sampling algorithm that is then coupled with a uniform sampling oracle; see Section 5 for details). Overall we show that, after preprocessing $\mathbf{A}$ with a randomized Hadamard transform, a uniform sub-sample of size k yields the same bound on the first moment of $\mathbf{P}$ as a DPP sample of size $\ell=\Omega(k/\log k)$, which closes the gap in the result of Dereziński and Yang (2024) from $O(\log^3 n)$ to $O(\log k)$. We note that standard lower bounds based on the coupon collector problem (Tropp, 2011) suggest that the factor $O(\log k)$ is unavoidable, meaning that our reduction size of $\Omega(k/\log k)$ is likely optimal.

Moreover, our first-moment analysis provides a lower bound on the entire matrix moment of $\mathbf{P}$, rather than just its smallest eigenvalue, which proves crucial for bounding ν later on. This result can be informally stated as follows: if $\mathbf{P}$ is produced from an $n\times n$ matrix $\mathbf{A}$

using a sub-sampling or sketching matrix $\mathbf{S}$ of size $k \times n$, then:

$$\text{(Lemma 12)}\quad \mathbb{E}[\mathbf{P}] \succeq_\delta \mathbf{A}^\top\mathbf{A}(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})^{-1}, \quad \text{where} \quad \lambda = \frac{1}{\ell}\sum_{i>\ell}\sigma_i^2 \quad \text{for} \quad \ell = \Omega\Big(\frac{k}{\log k}\Big),$$

where $\succeq_\delta$ hides an additive $\delta\mathbf{I}$. This result in particular implies a lower bound on μ, which roughly states that $\mu = \Omega(\ell/(n\kappa_\ell^2))$. As an immediate corollary of our new first-moment analysis of the random projection matrix $\mathbf{P}$, we can sharpen the complexity of solving linear systems with ℓ large singular values, i.e., those where $\kappa_\ell = O(1)$, the setting considered by Dereziński and Yang (2024). For comparison, they obtained $O((n^2\log^4 n + n\ell^{\omega-1}\log^{3\omega} n)\log 1/\epsilon)$ for this problem.

Corollary 5 *Consider an $n \times n$ matrix $\mathbf{A}$ with at most ℓ singular values larger than $O(1)$ times its smallest one, and vector $\mathbf{b}$. We can compute $\tilde{\mathbf{x}}$ such that $\|\mathbf{Ax}-\mathbf{b}\| \le \epsilon\|\mathbf{b}\|$ in:*

$$O\Big((n^2\log^2 n + n\ell^{\omega-1}\log^\omega \ell)\log 1/\epsilon\Big) \quad \textit{time.}$$

Technical contribution 2: *Second-moment projection analysis via Gaussian universality.* In our second main technical contribution, which is arguably the more significant of the two, we give the first non-trivial upper bound on the normalized second matrix moment of the random projection matrix $\mathbf{P}$. Here, our analysis crucially builds on recent breakthroughs in non-asymptotic random matrix theory (Brailovskaya and van Handel, 2024), which show that the singular values of certain classes of random matrices are close to the singular values of a Gaussian matrix with matching entry-wise mean and covariance (we refer to this property as *Gaussian universality*). In our case, Gaussian universality is required for the random sketch $\mathbf{SA}$, where $\mathbf{S}$ is a sparse sketching matrix. Fortunately, we are able to show that even extremely sparse sketching matrices (Dereziński et al., 2021; Chenakkod et al., 2024), for which computing $\mathbf{SA}$ costs only $\tilde{O}(nk)$, are sufficient for this to hold. We use this to show that when an $n\times n$ matrix $\mathbf{A}$ is preprocessed with a randomized Hadamard transform, then the resulting projection $\mathbf{P} = (\mathbf{SA})^\dagger\mathbf{SA}$ satisfies:

$$\text{(Lemma 20)}\quad \mathbb{E}\big[\mathbf{P}\bar{\mathbf{P}}^{-1}\mathbf{P}\big] \precsim \frac{n}{\ell}\cdot\bar{\mathbf{P}}, \quad \text{for} \quad \ell = \Omega\Big(\frac{k}{\log k}\Big),$$

where recall that $\bar{\mathbf{P}} = \mathbb{E}[\mathbf{P}]$ (we refer to Lemma 20 for a formal statement). Note that the result directly implies a bound on the largest eigenvalue of $\mathbb{E}[(\bar{\mathbf{P}}^{-1/2}\mathbf{P}\bar{\mathbf{P}}^{-1/2})^2]$, leading to $\nu \lesssim n/\ell$. Putting this together with the bound on μ, with a careful choice of the parameters we obtain $\sqrt{\mu/\nu} \gtrsim \ell/(n\kappa_\ell)$, which can be directly plugged into the convergence rate of Sketch-and-Project with Nesterov's acceleration (1).

The key reason we rely on Gaussian universality as part of the proof of Lemma 20 is to establish the following lower bound on the smallest singular value for $k \times n$ sketches $\mathbf{SA}$, which should be of independent interest:

$$\text{(Lemma 22)}\quad \sigma_{\min}^2(\mathbf{SA}) = \Omega\Big(2k\sigma_{2k}^2 + \sum_{i>2k}\sigma_i^2\Big) \quad \text{with high probability,}$$

where recall that $\sigma_1 \ge \sigma_2 \ge ...$ are the singular values of $\mathbf{A}$. Remarkably, this bound appears to be new even for dense Gaussian and sub-Gaussian sketching matrices $\mathbf{S}$, but crucially, we are able to show this also for the above mentioned sparse sketching matrices.

Completing the complexity analysis. To recover the time complexity in Theorem 1, we must also address the cost of the projection step in each iteration of the algorithm. This step effectively involves solving a very wide $k \times n$ linear system, which can be done inexactly in time $\tilde{O}(nk + k^\omega)$ by relying on standard sketch-and-precondition techniques (Rokhlin and Tygert, 2008; Meng et al., 2014; Woodruff, 2014). This forces us to extend the existing convergence analysis of Sketch-and-Project with Nesterov's acceleration (Gower et al., 2018) to allow inexact projections, which is done in Section 7. Putting everything together and choosing sketch size $k = \tilde{O}(\ell)$, our convergence analysis shows that the algorithm requires $\tilde{O}(\frac{n}{\ell}\kappa_\ell \cdot \log 1/\epsilon)$ iterations, each costing $\tilde{O}(n\ell + \ell^\omega)$ time. Setting $\ell = \tilde{O}(n^{\frac{1}{\omega-1}})$ yields the desired time complexity.

2. Related Work

Next, we give additional background and related work, focusing on different approaches to iteratively solving linear systems, as well as connections between our techniques and other areas such as matrix sketching, combinatorial sampling and random matrix theory.

Complexity of iterative linear solvers. A number of prior works have given sharp characterizations of the convergence of iterative linear system solvers, going beyond the usual condition number κ in some way, which makes them relevant to our discussion. In the context of Krylov methods such as CG, early works such as Axelsson and Lindskog (1986) showed that its convergence rate depends on the "clusterability" of the spectrum of $\mathbf{A}$, which has been restated in terms of κ_ℓ, as discussed earlier (see, e.g., Theorem 2.5 in Spielman and Woo, 2009).

For stochastic iterative methods, such as randomized Kaczmarz and randomized coordinate descent, a number of *averaged* notions of the condition number have been used (Strohmer and Vershynin, 2009; Leventhal and Lewis, 2010; Liu and Wright, 2016; Bollapragada et al., 2024), which depend in a more complex way on the entire spectrum, and may also lead to sharper bounds than κ. We recover similar averaged condition numbers in our analysis (Theorems 10 and 11). This has been used to suggest that stochastic methods can be faster than CG (Lee and Sidford, 2013), although without providing fine-grained upper/lower bounds, such as those given here based on the parameter ℓ. A few works have also considered versions of spectral tail condition numbers, similar to κ_ℓ, either as part of the analysis (Gonen et al., 2016; Musco et al., 2018b) or the assumptions (Dereziński and Yang, 2024). Some acceleration schemes other than Nesterov's momentum have been proposed for stochastic solvers (Lin et al., 2015; Frostig et al., 2015; Allen-Zhu, 2018; Bollapragada et al., 2024; Alderman et al., 2024), which can give sharper convergence guarantees in some cases, at the expense of additional computational overhead. Remarkably, we are able to show our results for Nesterov's original scheme, which is both simple and practical.

Matrix sketching and sampling. Randomized techniques for sketching or subsampling matrices have been developed largely as part of the area known as Randomized Numerical Linear Algebra (Woodruff, 2014; Drineas and Mahoney, 2016; Martinsson and Tropp, 2020; Dereziński and Mahoney, 2024), with applications in least squares (Sarlos, 2006; Rokhlin and Tygert, 2008) and low-rank approximation (Cohen et al., 2015, 2017), among others. In this context, the sketched projection matrix $\mathbf{P} = (\mathbf{SA})^\dagger\mathbf{SA}$, which we study as part of our analysis, arises naturally in low-rank approximation when bounding error of the form

$\|\mathbf{A} - \mathbf{AP}\|$, where $\mathbf{AP}$ is the projection of the input matrix $\mathbf{A}$ onto the rank k span of the sketch $\mathbf{SA}$ (Halko et al., 2011).

Some of the most popular and efficient sketching techniques are sparse random matrices, including CountSketch (Charikar et al., 2004), OSNAP (Nelson and Nguyên, 2013), and LESS (Dereziński et al., 2021). In our setting, we require sketching matrices that are even sparser than typically used, since they are applied repeatedly, so we rely on recently proposed LESS-uniform sketches (Dereziński et al., 2021; Chenakkod et al., 2024) in conjunction with the randomized Hadamard transform (Ailon and Chazelle, 2009; Tropp, 2011).

Most of the popular matrix sub-sampling methods are based on i.i.d. importance sampling, for example using the so-called leverage scores (Drineas et al., 2006, 2012). More recently, a number of works have used non-i.i.d. combinatorial sampling techniques, such as volume sampling and determinantal point processes, for solving matrix problems including least squares and low-rank approximation (for an overview, see Dereziński and Mahoney, 2021). We build on some of these approaches internally as part of our analysis (Section 5).

Sketch-and-Project. The sketch-and-project method was developed as a unified framework for iteratively solving linear systems (Gower and Richtárik, 2015a). Varying the distribution of the sketching matrix $\mathbf{S}$ and the projection type determined by a positive definite matrix $\mathbf{B}$, this general update rule reduces to a variety of classical methods such as randomized Gaussian pursuit, Randomized Kaczmarz, and Randomized Newton methods, among others (e.g., Gower et al., 2019; Gower and Richtárik, 2015a). The applicability of the sketch-and-project framework spans beyond linear solvers, including extensions aimed at solving linear and convex feasibility problems (Necoara et al., 2019), efficiently minimizing convex functions (Gower et al., 2019; Hanzely et al., 2020), solving nonlinear equations (Yuan et al., 2022) and inverting matrices (Gower and Richtárik, 2017).

The performance of generic sketch-and-project type methods is often hindered by the cost of storing and applying the sketching matrix, as well as the complexity of solving the sketched system (projection step), especially when the sketch size is not very small. Special features of the sketch-and-project based algorithm presented in this work (Algorithm 1)—such as ultra-sparse sketching matrices, momentum, and solving sketched systems iteratively—take care of the mentioned inefficiencies of the original sketch-and-project method and can be employed to refine the sketch-and-project methods beyond linear solvers; partial results in this direction were obtained in Hanzely et al. (2018); Gazagnadou et al. (2022); Dereziński et al. (2021); Dereziński and Yang (2024).

Random matrix theory. Due to the random nature of the sketching matrix $\mathbf{S}$, the analysis of Sketch-and-Project is greatly facilitated by advances in random matrix theory. In the asymptotic setting, when $\mathbf{S}$ has i.i.d. entries, the spectrum of $\mathbf{SA}$ converges deterministically to a generalized Marchenko–Pastur distribution (Silverstein and Bai, 1995). This spectral characterization enables a precise understanding of the projection matrix $\mathbf{P}$ and its expectation $\bar{\mathbf{P}}$ through deterministic equivalents in the asymptotic regime (Hachem et al., 2007; Rubio and Mestre, 2011), recently extended beyond the i.i.d. setting to asymptotically free sketches by LeJeune et al. (2024), who also applied this asymptotic analysis to Sketch-and-Project.

In finite dimensions, there are a variety of known results on matrix concentration (see Vershynin, 2018). In particular, the largest and smallest singular values of i.i.d. random matrices are well characterized, e.g., Rudelson and Vershynin (2009); Oymak and Tropp

(2018). For other types of random matrices, recent universality results (Brailovskaya and van Handel, 2024) prove that a broad class of sparse and non-i.i.d. matrices concentrate around the same spectral profile as Gaussian matrices, enabling the application of established results to new types of extremely sparse sketches (Chenakkod et al., 2024).

3. Preliminaries

Notation. We denote matrices with upper case bolded font ($\mathbf{X}$), vectors with lower case bolded font ($\mathbf{x}$), and scalars with basic font (x). The vector $\mathbf{e}_i$ denotes the ith coordinate vector. We denote by $[m]$ the set of integers from 1 to m, $[m] := \{1, 2, \dots m\}$. We denote expectation by $\mathbb{E}$ and probability by Pr, where the randomness should be clear from context if not specified. The smallest and largest (non-zero) singular values of a matrix $\mathbf{X}$ are denoted by $\sigma_{\min}(\mathbf{X})$ and $\sigma_{\max}(\mathbf{X})$, respectively. The norm $\|\cdot\|$ denotes the Euclidean or spectral norm. The condition number of a matrix $\mathbf{X}$ is denoted by $\kappa(\mathbf{X}) = \sigma_{\max}(\mathbf{X})\sigma_{\min}^{-1}(\mathbf{X})$. For a symmetric positive definite matrix $\mathbf{A}$, its induced norm is defined as $\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^\top \mathbf{A}\mathbf{x}}$, and we write $\mathbf{A} \preceq \mathbf{B}$ to denote the (Loewner) matrix ordering, i.e., to say that $\mathbf{B} - \mathbf{A}$ is positive semi-definite. For an event $\mathcal{E}$, we write $\neg\mathcal{E}$ to denote the complement of $\mathcal{E}$. Lastly, we utilize constants $c, c_1, c_2, \dots, C, C_1, C_2, \dots$ to denote absolute constants, which may hold different values from one instance to the next. We reserve the use of big-O notation for complexity remarks, and we use the notation $\tilde{O}$ to hide logarithmic factors. To avoid difficulty in implicitly adjusting constants, we assume that m, n, k, ℓ, δ are bounded away from 1.

Sketching model. We obtain fast solvers for our main results with Algorithm 1, which is based on the Sketch-and-Project framework (Gower and Richtárik, 2015a), with ultra-sparse sketching matrices $\mathbf{S}$ and Nesterov's acceleration. For ultra-sparse sketching, we consider the following sparse sketching construction, following similar constructions in Dereziński et al. (2021); Chenakkod et al. (2024).

Definition 6 (LESS-uniform) *A sparse sketching matrix $\mathbf{S} \in \mathbb{R}^{k\times m}$ is called a LESS-uniform matrix with s non-zeros per row if it is constructed out of independent and identically distributed (i.i.d.) rows, with the ith row of $\mathbf{S}$ given by*

$$\sqrt{\frac{m}{s}} \sum_{j=1}^{s} r_{ij} \mathbf{e}_{I_{ij}}^\top,$$

where $I_{i1}, ..., I_{is}$ are i.i.d. random indices sampled uniformly with replacement from $[m]$, and r_{ij} are i.i.d. Rademacher random variables, i.e., $\Pr(r_{ij} = 1) = \Pr(r_{ij} = -1) = 1/2$.

This is one of the most natural ways to generate sparse sketching matrices. We can avoid more expensive techniques used in earlier works (Dereziński and Rebrova, 2024) thanks to initial preprocessing with the randomized Hadamard transform (Ailon and Chazelle, 2009).

Definition 7 (Randomized Hadamard Transform) *For any m that is a power of 2, the Hadamard matrix $\mathbf{H}_m \in \mathbb{R}^{m\times m}$ is defined so that $\mathbf{H}_0 = 1$ and:*

$$\mathbf{H}_m = \begin{bmatrix} \mathbf{H}_{m/2} & \mathbf{H}_{m/2} \\ \mathbf{H}_{m/2} & -\mathbf{H}_{m/2} \end{bmatrix}.$$

An $m \times m$ randomized Hadamard transform (RHT) is a random matrix $\mathbf{Q} = \frac{1}{\sqrt{m}}\mathbf{H}_m\mathbf{D}$, where $\mathbf{H}_m$ is the Hadamard matrix and $\mathbf{D}$ is an $m \times m$ diagonal matrix with i.i.d. Rademacher entries. Applying $\mathbf{Q}$ to a vector takes $O(m \log m)$ time.

An advantage of the RHT is that it can be used to uniformize the leverage scores of a matrix, which is crucial in our analysis of both the first and second matrix moments of the random projection $\mathbf{P}$.

Lemma 8 (Lemma 3.3, Tropp (2011)) *Consider a matrix $\mathbf{U} \in \mathbb{R}^{m\times d}$ such that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ and let $\mathbf{Q}$ be the $m \times m$ Randomized Hadamard Transform. Then, the matrix $\tilde{\mathbf{U}} = \mathbf{Q}\mathbf{U}$ with probability $1 - \delta$ has nearly uniform leverage scores, i.e., the norms of its rows are bounded as $\|\tilde{\mathbf{u}}_i\| \leq \sqrt{d/m} + \sqrt{8\log(m/\delta))/m}$ for all $i = 1, \ldots, m$.*

As an auxiliary result, we obtain new bounds on the extreme singular values of the random matrix $\mathbf{SU}$, where $\mathbf{S}$ is a $k \times m$ LESS-uniform embedding and $\mathbf{U}$ is an $m \times d$ isometric embedding matrix, i.e., such that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$. Specifically, in Appendix A we show that $\mathbf{SU}$ matrices have extreme singular values of the same order as after Gaussian sketching.

Lemma 9 *Consider an $m \times d$ matrix $\mathbf{U}$ such that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ and such that the norm of each row of $\mathbf{U}$ is bounded by $C\sqrt{d/m}$. There are universal constants $c_1, c_2, C_1, C_2 > 0$ such that if $\mathbf{S}$ is a $k \times m$ LESS-uniform matrix with $1 \leq k \leq d/2$, $s \geq C_1 \log^4(d/\delta)$ non-zeros per row, and dimension $d \geq C_2 \log(1/\delta)$, then with probability $1 - \delta$*

$$c_1\sqrt{d} \leq \sigma_{\min}(\mathbf{SU}) \leq \sigma_{\max}(\mathbf{SU}) \leq c_2\sqrt{d}.$$

We use the upper bound from Lemma 9 in our first-moment projection analysis (Section 5), while the lower bound is needed for the second-moment analysis (Section 6). Crucially, a similar result does not hold if we replace the LESS-uniform matrix with a uniform sub-sampling matrix (the smallest singular value of a uniformly sub-sampled $\mathbf{U}$ can be arbitrarily close to zero with a non-negligible probability when the sample size k is less than d).

Numerical stability. Even though, for simplicity, our results are stated in the exact arithmetic (Real-RAM) model, we provide a stability analysis of our algorithm in Section 7, which shows that its convergence rate is stable with respect to the error in the projection steps. Extending this analysis to the Word-RAM model with word sizes polylogarithmic in the parameters of the problem is a promising direction, since all other linear algebraic operations we rely on are known to be stable (Demmel et al., 2007). This suggests another potential advantage of Sketch-and-Project type solvers over deterministic Krylov methods, for which ensuring stability is an ongoing area of research (Musco et al., 2018a; Peng and Vempala, 2021).

4. Main Algorithm and Convergence Guarantees

Our main results consist of convergence theorems describing the iteration-wise convergence rate of Algorithm 1, revealing sharp dependencies on the spectral profile of matrix $\mathbf{A}$. Here, note that the algorithm takes as an additional parameter positive definite matrix $\mathbf{B}$, which controls the type of projection operator used in Sketch-and-Project.

We first present the more general convergence result for an $m \times n$ matrix $\mathbf{A}$, where for simplicity we assume that $m \geq n$ and $\mathbf{A}$ has full column rank. To fully capture the

Algorithm 1 Sketch-and-Project with Nesterov's acceleration

```
Input: matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, vector $\mathbf{b} \in \mathbb{R}^m$, p.d. matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, sketch size $k$, iterate $\mathbf{x}_0$, iteration $t_{\max}$, $\beta = 1 - \sqrt{\frac{\mu}{\nu}}$, $\gamma = \frac{1}{\sqrt{\mu\nu}}$, $\alpha = \frac{1}{1+\gamma\nu}$;
$\mathbf{v}_0 \leftarrow \mathbf{x}_0$;
for $t = 0, 1, \ldots, (t_{\max} - 1)$ do
    $\mathbf{y}_t \leftarrow \alpha\mathbf{v}_t + (1-\alpha)\mathbf{x}_t$;
    Generate sketching matrix $\mathbf{S}$;
    Solve $(\mathbf{S}\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}^\top)\mathbf{u}_t = \mathbf{S}\mathbf{A}\mathbf{y}_t - \mathbf{S}\mathbf{b}$ for $\mathbf{u}_t$      ▷ Possibly inexactly, see Section 7
    $\mathbf{w}_t \leftarrow \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}^\top\mathbf{u}_t$;
    $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t - \mathbf{w}_t$;      ▷ $\mathbf{x}_{t+1} = \mathrm{argmin}_{\mathbf{x}}\|\mathbf{x} - \mathbf{y}_t\|_{\mathbf{B}}$ s.t. $\mathbf{S}\mathbf{A}\mathbf{x} = \mathbf{S}\mathbf{b}$
    $\mathbf{v}_{t+1} \leftarrow \beta\mathbf{v}_t + (1-\beta)\mathbf{y}_t - \gamma\mathbf{w}_t$;
end for
return $\tilde{\mathbf{x}} = \mathbf{x}_{t_{\max}}$;      ▷ Solves $\mathbf{A}\mathbf{x} = \mathbf{b}$.
```

convergence properties in terms of the spectral profile of $\mathbf{A}$, we rely on the following notion of average condition number, $\bar{\kappa}_{\ell:q}$, parameterized by two indices $1 \leq \ell < q \leq n$ describing the range of singular values on which the condition number is computed:

$$\bar{\kappa}_{\ell:q} = \Big(\frac{1}{q-\ell}\sum_{i=\ell+1}^{q}\frac{\sigma_i^2}{\sigma_q^2}\Big)^{1/2},$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$ are the singular values of $\mathbf{A}$. We use $\bar{\kappa}_\ell := \bar{\kappa}_{\ell:n}$ as a shorthand.

Theorem 10 *Consider an $m \times n$ matrix $\mathbf{A}$ with full column rank, and a vector $\mathbf{b}$ such that the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is consistent. Suppose that we preprocess the linear system with a randomized Hadamard transform $\mathbf{Q}$ by setting $\mathbf{A} \leftarrow \mathbf{Q}\mathbf{A}$ and $\mathbf{b} \leftarrow \mathbf{Q}\mathbf{b}$, and choose $\mathbf{B} = \mathbf{I}$ in Algorithm 1. Also, let the sketching matrices $\mathbf{S}$ be LESS-uniform with sketch size $k \geq C_1\log(m/\delta)$, $2k \leq m$ and $s \geq C_2\log^4(n/\delta)$ non-zeros per row. Then, conditioned on a $1-\delta$ probability event that only depends on the randomness of $\mathbf{Q}$,*

$$\mathbb{E}\big[\|\mathbf{x}_t - \mathbf{x}^*\|^2\big] \leq 2\Big(1 - \frac{c_1\ell}{n\bar{\kappa}_\ell\bar{\kappa}_{\ell:2k}}\Big)^t\|\mathbf{x}_0 - \mathbf{x}^*\|^2 \qquad \text{for} \qquad \ell = \Big\lceil\frac{c_2k}{\log k}\Big\rceil,$$

where $\mathbf{x}^*$ is the minimum norm solution. Moreover, the algorithm can be implemented to run in time $O(mn\log m + t(nk(s + \log(n\bar{\kappa}_\ell)) + k^\omega))$.

Remark 2 *While our results are stated for $\mathbf{A}$ with full column rank, they should extend naturally to rank-deficient $\mathbf{A}$ by operating on the subspace defined by the column span of $\mathbf{A}$, following arguments similar to those of Gower and Richtárik (2015b). We focus here on the full-rank setting for the sake of better clarity of the notation and analysis.*

In the case where $\mathbf{A}$ is positive definite, the additional structure provides us with an improved condition number dependence in the convergence rate, by a square root factor. This is achieved by using a modified projection operator dictated by the choice of $\mathbf{B}$. Here, the average condition number is slightly redefined in terms of the eigenvalues of $\mathbf{A}$ (instead of its singular values) to suite the positive definite setting:

$$\tilde{\kappa}_{\ell:q} = \frac{1}{q-\ell}\sum_{i=\ell+1}^{q}\frac{\lambda_i}{\lambda_q},$$

where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n > 0$ are the eigenvalues of the $n \times n$ positive definite $\mathbf{A}$. Here, similarly as before, we use $\tilde{\kappa}_\ell$ as a shorthand for $\tilde{\kappa}_{\ell:n}$.

Theorem 11 *Consider an $n \times n$ positive definite matrix $\mathbf{A}$ and a vector $\mathbf{b} \in \mathbb{R}^n$. Suppose that we initialize Algorithm 1 by setting both $\mathbf{A}$ and $\mathbf{B}$ to $\mathbf{QAQ}^\top$ and replacing $\mathbf{b}$ with $\mathbf{Qb}$. Also, let the sketching matrices $\mathbf{S}$ be LESS-uniform with sketch size $k \geq C_1 \log(n/\delta)$, $2k \leq n$ and $s \geq C_2 \log^4(n/\delta)$ non-zeros per row. Then, conditioned on a $1-\delta$ probability event that only depends on the randomized Hadamard transform $\mathbf{Q}$,*

$$\mathbb{E}\big[\|\mathbf{x}_t - \mathbf{x}^*\|_{\mathbf{A}}^2\big] \leq 2\Big(1 - \frac{c_1\ell}{n\sqrt{\tilde{\kappa}_\ell \tilde{\kappa}_{\ell:2k}}}\Big)^t \|\mathbf{x}_0 - \mathbf{x}^*\|_{\mathbf{A}}^2 \qquad for \qquad \ell = \Big\lceil \frac{c_2 k}{\log k}\Big\rceil,$$

with $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$ denoting the solution. Moreover, the algorithm can be implemented to run in time $O(n^2 \log n + t(nks + k^\omega))$.

We next briefly discuss how the time complexities given in Theorem 1 and 2 can be recovered from the above convergence results, with the details relegated to Section 8.

Following the assumptions of Theorem 1, suppose that we are given an $n \times n$ invertible matrix $\mathbf{A}$ and an n-dimensional vector $\mathbf{b}$. If the sketch size in Algorithm 1 satisfies $k = \tilde{O}(n^{\frac{1}{\omega-1}})$ and $\ell = \Omega(k/\log k)$, then we can write down the time complexity described by Theorem 10 as follows:

$$\tilde{O}\Big(n^2 + \bar{\kappa}_\ell \bar{\kappa}_{\ell:2k}\frac{n}{\ell}(nk + k^\omega)\Big) = \tilde{O}\Big(\bar{\kappa}_\ell \bar{\kappa}_{\ell:2k}(n^2 + nk^{\omega-1})\Big) = \tilde{O}\big(\bar{\kappa}_\ell \bar{\kappa}_{\ell:2k} n^2\big),$$

and an analogous calculation holds for Theorem 11, with $\bar{\kappa}_\ell \bar{\kappa}_{\ell:2k}$ replaced by $\sqrt{\tilde{\kappa}_\ell \tilde{\kappa}_{\ell:2k}}$. It remains to translate those condition number quantities into our spectral tail condition number $\kappa_\ell = \frac{\sigma_\ell}{\sigma_n}$. Fortunately, in Section 8 we show that one can always find a sketch size $k = \tilde{O}(n^{\frac{1}{\omega-1}})$ and the corresponding $\ell = \tilde{\Omega}(k)$, such that those quantities are bounded by $\tilde{O}(\kappa_\ell)$ and $\tilde{O}(\sqrt{\kappa_\ell})$, respectively, which is sufficient to establish Theorems 1 and 2. In the next sections, we present the convergence analysis of Sketch-and-Project with Nesterov's acceleration, needed for Theorems 10 and 11, which is our main technical contribution.

5. First-Moment Projection Analysis via Optimal DPP Reduction

In this section, we show how to lower bound the smallest eigenvalue of the expected projection matrix, i.e., $\mu = \lambda_{\min}(\mathbb{E}[\mathbf{P}])$, where $\mathbf{P} = (\mathbf{SA})^\dagger \mathbf{SA}$ is the random projection matrix central to the convergence guarantee (1) for Sketch-and-Project. In fact, we give a more general result, lower bounding the entire expected projection in positive semidefinite ordering, which will be necessary later for the analysis of the second-order term ν.

Lemma 12 *Suppose that a rank n matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is transformed by the randomized Hadamard transform (RHT), i.e., $\mathbf{A} \leftarrow \mathbf{QA}$. If $\mathbf{S}$ is a $k \times m$ LESS-uniform sketching matrix with sketch size $k \in [C_1 \log(m/\delta), n]$ and s non-zeros per row, then, as long as $s = O(1)$ or $s \geq C_2 \log^4(n/\delta)$, and $n \geq C_3 \log(1/\delta)$, conditioned on an RHT property that holds with probability $1-\delta$, there exists $\ell \geq \frac{ck}{\log k}$ such that the matrix $\mathbf{P} = (\mathbf{SA})^\dagger \mathbf{SA}$ satisfies:*

$$\mathbb{E}[\mathbf{P}] \succeq \mathbf{A}^\top \mathbf{A}(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I})^{-1} - \delta \mathbf{I}, \qquad \textit{where} \qquad \lambda = \frac{1}{\ell}\sum_{i>\ell}\sigma_i^2.$$

First, we show how the above result can be translated into a lower bound on the smallest eigenvalue of the expected projection, and therefore, on μ.

Corollary 13 *Under the assumptions of Lemma 12, and choosing $\delta \leq \frac{1}{n\bar{\kappa}_\ell^2}$, we have*

$$\lambda_{\min}(\mathbb{E}[\mathbf{P}]) \geq \frac{\ell/4}{n\bar{\kappa}_\ell^2} \quad \textit{for} \quad \ell = \left\lceil \frac{ck}{\log k}\right\rceil, \quad \bar{\kappa}_\ell = \Big(\frac{1}{n-\ell}\sum_{i>\ell}\frac{\sigma_i^2}{\sigma_n^2}\Big)^{1/2}.$$

Proof Note that we can choose a sufficiently large C_1 in the lemma to make sure that k is large enough so that $\ell = \lceil \frac{ck}{\log k}\rceil \in [4, n/2]$. Thus, we have $\lambda = \frac{1}{\ell}\sum_{i>\ell}\sigma_i^2 \geq \frac{1}{n-\ell}\sum_{i>\ell}\sigma_i^2 \geq \sigma_n^2$. Note that if the lower bound of Lemma 12 holds with some $\ell \geq \frac{ck}{\log k}$, then it holds with $\ell = \lceil \frac{ck}{\log k}\rceil$. Thus, using Lemma 12 with $\delta \leq \frac{1}{n\bar{\kappa}_\ell^2}$, we have:

$$\lambda_{\min}(\mathbb{E}[\mathbf{P}]) \geq \lambda_{\min}(\mathbf{A}^\top\mathbf{A}(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})^{-1}) - \delta = \frac{\sigma_n^2}{\sigma_n^2+\lambda} - \delta \geq \frac{\sigma_n^2}{2\lambda} - \delta = \frac{\ell/2}{(n-\ell)\bar{\kappa}_l^2} - \delta \geq \frac{\ell/4}{n\bar{\kappa}_\ell^2},$$

which concludes the proof of the corollary. ■

Remark 3 *We note that for the case of $s = 1$ non-zeros per row, the LESS-uniform sketch is precisely equivalent to uniform sub-sampling. In this case, Corollary 13 is a direct improvement over the result of Dereziński and Yang (2024), from $\ell = \Omega(\frac{k}{\log^3 m})$ to $\ell = \Omega(\frac{k}{\log k})$ (their Lemma 4.2).*

Before we proceed with the proof of Lemma 12, we introduce some definitions related to determinantal point processes, a family of non-i.i.d. sub-sampling distributions commonly studied in machine learning (Kulesza and Taskar, 2012), that are central to our analysis.

Definition 14 *Given an $m \times m$ positive semi-definite matrix $\mathbf{B}$, a determinantal point process $\mathcal{S} \sim \mathrm{DPP}(\mathbf{B})$ is a distribution over all sets $\mathcal{S} \subseteq \{1,...,m\}$ such that $\Pr(\mathcal{S}) \propto \det(\mathbf{B}_{\mathcal{S},\mathcal{S}})$, where $\mathbf{B}_{\mathcal{S},\mathcal{S}}$ denotes the prinicipal submatrix of $\mathbf{B}$ indexed by $\mathcal{S}$.*

The distribution of $\mathcal{S} \sim \mathrm{DPP}(\mathbf{B})$ is defined over all 2^m subsets of $\{1,...,m\}$, which means that the size of $\mathcal{S}$ is also a random variable (albeit one can show that it is highly concentrated around its mean), hence we call this a random-sized DPP. If we condition the DPP on a particular set size, $|\mathcal{S}| = k$, the resulting distribution is often called a (fixed-size) k-DPP.

The key benefit of determinantal point processes in the context of Sketch-and-Project is that, thanks to a Cauchy-Binet-type determinantal summation formula, the expectation of the random projection matrix $\mathbf{P}$ has a simple closed form under DPP sampling, which, as we see below, is needed for our analysis.

Lemma 15 (Lemma 5, Dereziński et al. (2020)) *Given matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, let $\mathcal{S} \sim \mathrm{DPP}(\frac{1}{\lambda}\mathbf{A}\mathbf{A}^\top)$ for some $\lambda > 0$, and define $\mathbf{S} \in \mathbb{R}^{|\mathcal{S}|\times m}$ to be the sampling matrix corresponding to $\mathcal{S}$, i.e., a matrix consisting of rows that are standard basis vectors $\mathbf{e}_i^\top$ for $i \in \mathcal{S}$, so that $\mathbf{SA}$ is the submatrix of the rows of $\mathbf{A}$ indexed by $\mathcal{S}$. Then, we have*

$$\mathbb{E}[(\mathbf{SA})^\dagger\mathbf{SA}] = \mathbf{A}^\top\mathbf{A}(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I})^{-1}. \tag{2}$$

If we could use a DPP sampling matrix instead of our LESS-uniform sketching matrix in the statement of Lemma 12, then the desired lower bound on the expectation of $\mathbf{P}$ would follow directly from this result. However, unfortunately, we cannot rely on sampling directly from a DPP for two reasons. First, despite considerable work in this direction (see Dereziński and Mahoney, 2021, for an overview), the cost of sampling from these combinatorial distributions far exceeds the time complexity of uniform sub-sampling, or of our LESS-uniform sparse sketching, which is necessary to recover our main results. Second, while DPPs provide a closed form for the first moment of the projection $\mathbf{P}$, the same is not true for the normalized second moment, required to bound ν in (1). To achieve this second guarantee, we must rely on Gaussian universality properties which are specific to sketching matrices such as LESS-uniform.

Our proof of Lemma 12 starts with the observation that we do not actually need to sample from a DPP to produce a lower bound on $\mathbb{E}[\mathbf{P}]$, but rather, it suffices to produce a sample that contains a DPP sample. This is true because for any two subsets $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq [m]$ and the corresponding sampling matrices $\mathbf{S}_1$, $\mathbf{S}_2$, the projection matrices $\mathbf{P}_{\mathcal{S}_i} := (\mathbf{S}_i\mathbf{A})^\dagger\mathbf{S}_i\mathbf{A}$ satisfy $\mathbf{P}_{\mathcal{S}_1} \preceq \mathbf{P}_{\mathcal{S}_2}$. This leads to the following strategy: construct a hypothetical sampling algorithm which produces a DPP sample $\mathcal{S}_{\mathrm{DPP}}$, but also as a byproduct, defines another sample $\mathcal{S}$ such that: (a) the sample $\mathcal{S}$ contains the DPP sample, i.e., $\mathcal{S}_{\mathrm{DPP}} \subseteq \mathcal{S}$, while not being too much larger; and (b) the distribution of $\mathcal{S}$ is easier to sample from (for example, uniform sampling). This strategy, which can be viewed as coupling together the distributions of $\mathcal{S}_{\mathrm{DPP}}$ and $\mathcal{S}$, is then used to show that $\mathbb{E}[\mathbf{P}_{\mathcal{S}}] \succeq \mathbb{E}[\mathbf{P}_{\mathcal{S}_{\mathrm{DPP}}}]$. Finally, we rely on the DPP expectation formula from Lemma 15 to complete the analysis, showing that the easy sampling scheme $\mathcal{S}$ enjoys the desired guarantee (without needing to implement the DPP algorithm).

Our coupling argument is based on a combination of two algorithms. The first one, Algorithm 2, is a classical method for sampling from a determinantal point process by reformulating it as a mixture of so-called Projection DPPs (Kulesza and Taskar, 2012), which are a special instance of a fixed-size k-DPP where k is the rank of the given matrix.

Definition 16 *Given an $m \times m$ positive semi-definite matrix $\mathbf{B}$, we define $\mathcal{S} \sim$P-DPP$(\mathbf{B})$ as a distribution over subsets $\mathcal{S} \subseteq \{1, ..., m\}$ of size $\mathrm{rank}(\mathbf{B})$ such that $\Pr(\mathcal{S}) \propto \det(\mathbf{B}_{\mathcal{S},\mathcal{S}})$.*

At a high level, Algorithm 2 first samples a subset of the left singular vectors of the input matrix $\mathbf{A}$ via independent Bernoulli coin flips, and then samples from a Projection DPP defined using just those vectors. The fact that this algorithm returns a sample from

Algorithm 2 Random-sized DPP sampler (Algorithm 1 from Kulesza and Taskar (2012))

```
input: $\mathbf{A} \in \mathbb{R}^{m\times n}$ with SVD $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ where $\mathbf{D} = \mathrm{diag}(\sigma_1, ..., \sigma_n)$
output: $\mathcal{S}_{\mathrm{DPP}} \sim \mathrm{DPP}(\mathbf{A}\mathbf{A}^\top)$
For each $i = 1, ..., n$, independently sample $b_i \sim \mathrm{Bernoulli}(\frac{\sigma_i^2}{\sigma_i^2+1})$
Construct matrix $\mathbf{U}_{*,\mathbf{b}}$ consisting of columns of $\mathbf{U}$ indexed by $\mathbf{b} = [b_1, ..., b_n]$
return $\mathcal{S}_{\mathrm{DPP}} \sim \mathrm{P\text{-}DPP}(\mathbf{U}_{*,\mathbf{b}}\mathbf{U}_{*,\mathbf{b}}^\top)$ using Algorithm 3
```

DPP($\mathbf{A}\mathbf{A}^\top$), while certainly non-trivial, has been known since the work of Hough et al. (2006). Note that, in particular, this scheme characterizes the distribution of the size of $\mathcal{S}_{\mathrm{DPP}}$, since this size is equal to the number of singular vectors that are selected.

Observe that Algorithm 2 technically requires an exact singular value decomposition (SVD) of the input matrix $\mathbf{A}$, which means that it is clearly far too expensive for our purposes. However, as explained above, we will only use it in our proof of Lemma 12.

We combine this algorithm with a recently proposed rejection sampling scheme for sampling a Projection DPP (Dereziński et al., 2019), given in Algorithm 3. This algorithm describes how we can select a P-DPP sample out of a stream of i.i.d. samples drawn according to a distribution q over $\{1, ..., m\}$. What matters for our coupling argument is the sample complexity of this procedure, i.e., how many i.i.d. samples from q need to be drawn by the algorithm (in line 6) to extract one P-DPP sample $\mathcal{S}_{\mathrm{DPP}}$ via rejection sampling. This stream of i.i.d. samples (including the rejected ones), which the algorithm collects into the set $\mathcal{S}$ (line 7), is going to be the larger sample that we couple with the DPP.

Algorithm 3 Projection DPP sampler

```
 1: input: $\mathbf{U} \in \mathbb{R}^{m\times \ell}$ such that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, probability distribution $q$ over $\{1, ..., m\}$ such
    that $\|\mathbf{u}_i\|^2 \le R\ell q_i$ for all $1 \le i \le m$ and some $R \ge 1$, where $\mathbf{u}_i^\top$ is the $i$th row of $\mathbf{U}$
 2: output: $\mathcal{S}_{\mathrm{DPP}} \sim \mathrm{P\text{-}DPP}(\mathbf{U}\mathbf{U}^\top)$
 3: Initialize $\mathbf{Z} = \mathbf{I}_\ell$, $\mathcal{S}_{\mathrm{DPP}} = \{\}$, $\mathcal{S} = \{\}$
 4: for $j = 1, ..., \ell$
 5:   repeat
 6:     Sample index $I \sim q$
 7:     $\mathcal{S} \leftarrow \mathcal{S} \cup \{I\}$
 8:     Sample  Accept $\sim \mathrm{Bernoulli}\big(\frac{\mathbf{u}_I^\top \mathbf{Z}\mathbf{u}_I}{R\ell q_I}\big)$
 9:   until Accept = 1
10:   $\mathcal{S}_{\mathrm{DPP}} \leftarrow \mathcal{S}_{\mathrm{DPP}} \cup \{I\}$
11:   $\mathbf{Z} \leftarrow \mathbf{Z} - \frac{\mathbf{Z}\mathbf{u}_I\mathbf{u}_I^\top\mathbf{Z}}{\mathbf{u}_I^\top\mathbf{Z}\mathbf{u}_I}$
12: end for
13: return $\mathcal{S}_{\mathrm{DPP}}$
```

To ensure that the rejection sampling used in Algorithm 3 terminates efficiently, we rely on the following prior result, which bounds the number of rejection sampling steps required across all iterations of the sampler.

Lemma 17 (Dereziński et al. (2019)) *Given $\mathbf{U} \in \mathbb{R}^{m\times\ell}$ such that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, let $\mathbf{u}_i^\top$ denote its ith row. Consider a probability distribution q over $\{1,...,m\}$ such that $\|\mathbf{u}_i\|^2 \leq R\ell q_i$ for all i and some $R \geq 1$. Then, Algorithm 3 produces a sample $\mathcal{S}_{\mathrm{DPP}} \sim \text{P-DPP}(\mathbf{U}\mathbf{U}^\top)$, and with probability $1-\delta$ it requires at most $|\mathcal{S}| = O(R\ell\log(\ell/\delta))$ samples drawn from q (line 6).*

The number of samples required by Algorithm 3 depends on R, which specifies how well q approximates the so-called leverage score sampling distribution of $\mathbf{U}$. For a matrix $\mathbf{U}$ such that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, its ith leverage score is defined as the norm squared of the ith row $\mathbf{u}_i$ of $\mathbf{U}$. In particular, if $R = 1$ in Lemma 17, then q_i corresponds to exact leverage score sampling, $q_i = \|\mathbf{u}_i\|^2/\sum_i \|\mathbf{u}_i\|^2 = \|\mathbf{u}_i\|^2/\ell$. This distribution effectively describes the likelihood of an index i being sampled into $\mathcal{S}_{\mathrm{DPP}}$, via the following formula: $\Pr(i \in \mathcal{S}_{\mathrm{DPP}}) = \|\mathbf{u}_i\|^2$.

Naturally, we cannot simply use $R = 1$ in Algorithm 3, because that would force q to be the exact leverage score sampling distribution, which is itself as expensive to compute as the SVD. Instead, in our algorithm, R is controlled by applying the RHT to the data matrix and relying on Lemma 8 to ensure that the row norms (and thereby the leverage scores) are uniformly bounded. This way, even a uniform sampling distribution q provides a decent enough approximation of leverage score sampling so that $R = \tilde{O}(1)$.

To further refine the coupling argument, and adapt it to LESS-uniform sketching, our proof of Lemma 12 also relies on the notion of total variation distance between two probability distributions, which we define here.

Definition 18 (Total variation distance) *Let μ, ν be probability measures defined on a measurable space $(\Omega, \mathcal{F})$, the total variation distance between μ and ν is defined as*

$$d_{\mathrm{tv}}(\mu,\nu) = \sup_{A\in\mathcal{F}} |\mu(A) - \nu(A)|.$$

Bounding the total variation distance allows us to couple two random variables so that they are equal with high probability.

Lemma 19 (Example 4.14, Van Handel (2014)) *Let μ, ν be probability measures defined on some $(\Omega, \mathcal{F})$. Then, we have $\inf_{(X,Y)} \Pr\{X \neq Y\} = d_{\mathrm{tv}}(\mu,\nu)$, where $X \sim \mu$, $Y \sim \nu$, and the infimum is taken over all couplings of μ and ν.*

We are now ready to give the proof of Lemma 12 by coupling a LESS-uniform sketch with a DPP sample produced by Algorithm 2 for a specially expanded version of matrix $\mathbf{A}$.

Proof of Lemma 12 As mentioned above, the general approach of our proof is to show that the LESS-uniform sketching matrix $\mathbf{S}$ defined in Lemma 12 essentially *contains* a DPP sample, so that we can rely on the simple closed form expression for the expected projection under DPP sampling (Lemma 15). To do this, we must first represent a LESS-unifom sketching matrix itself as a sample. We achieve this by defining an expanded matrix $\bar{\mathbf{A}}$, which contains all possible rows that could be produced via this sketching method.

Define the LESS-uniform *expansion* matrix $\bar{\mathbf{S}} \in \mathbb{R}^{N\times m}$, where $N = (2m)^s$ and s is the number of non-zeros per row in the definition of LESS-uniform (Definition 6), so that $\bar{\mathbf{S}}$ consists of all rows of the form:

$$\sqrt{\frac{m}{sN}}\sum_{i=1}^s r_i\mathbf{e}_{I_i}^\top, \quad r_1,...,r_s \in \{1,-1\}, \quad I_1,...,I_s \in [m].$$

Note that we have $\bar{\mathbf{S}}^\top\bar{\mathbf{S}} = \mathbf{I}$, because the LESS-uniform sketching distribution is isotropic. Now, let $\mathbf{U}\mathbf{D}\mathbf{V}^\top$ be the SVD of $\mathbf{A}$ and consider matrix $\bar{\mathbf{A}} = \bar{\mathbf{S}}\mathbf{Q}\mathbf{A}$, where recall that $\mathbf{Q}$ is the RHT matrix and it satisfies $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$. We can now write the SVD of $\bar{\mathbf{A}}$ as follows: $\bar{\mathbf{A}} = (\bar{\mathbf{S}}\tilde{\mathbf{U}})\mathbf{D}\mathbf{V}^\top$ where $\tilde{\mathbf{U}} = \mathbf{Q}\mathbf{U}$. Observe that we designed matrix $\bar{\mathbf{A}}$ so that an i.i.d. uniform sample of rows from $\bar{\mathbf{A}}$ is distributed identically to a LESS-uniform sketch of matrix $\mathbf{Q}\mathbf{A}$.

Next, we consider running Algorithm 2 on the expanded matrix to produce a sample $\mathcal{S}_{\mathrm{DPP}} \sim \mathrm{DPP}(\frac{1}{\lambda}\bar{\mathbf{A}}\bar{\mathbf{A}}^\top)$. The size of the sample is random, and it is equal to $\sum_i b_i$, since Algorithm 3 always selects one element per each $b_i = 1$. Thus, we can obtain the expected size of the DPP sample as follows, where σ_i's denote the singular values of $\mathbf{A}$:

$$\mathbb{E}\big[|\mathcal{S}_{\mathrm{DPP}}|\big] = \sum_{i=1}^n \mathbb{E}[b_i] = \sum_{i=1}^n \frac{\sigma_i^2/\lambda}{\sigma_i^2/\lambda + 1} = \sum_{i=1}^n \frac{\sigma_i^2}{\sigma_i^2 + \lambda}$$

Recall that, due to our choice of $\lambda = \frac{1}{\ell}\sum_{i>\ell}\sigma_i^2$, we have:

$$\sum_{i=1}^n \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \le \ell + \frac{1}{\lambda}\sum_{i>\ell}\sigma_i^2 = 2\ell.$$

From now on, we will define $\tilde{\ell} := |\mathcal{S}_{\mathrm{DPP}}| = \sum_i b_i$ as a shorthand for the random DPP sample size. Ultimately, our goal is to show that the sampler with high probability produces its output sample $\mathcal{S}_{\mathrm{DPP}}$ after drawing only at most $O(\ell\log\ell)$ uniform samples in line 6 of Algorithm 3, i.e., to bound the size of the set $\mathcal{S}$ produced in the course of the algorithm. The collection of rows of $\bar{\mathbf{A}}$ indexed by $\mathcal{S}$ forms a LESS-uniform sketch $\mathbf{S}\mathbf{Q}\mathbf{A}$, which allows us to argue that a $k = O(\ell\log\ell)$ size LESS-uniform sketch contains a $\mathrm{DPP}(\frac{1}{\lambda}\bar{\mathbf{A}}\bar{\mathbf{A}}^\top)$ sample.

We next define the high probability event depending only on the RHT, which will be used to ensure that the parameter R in Lemma 17 and Algorithm 3 can be bounded. First, consider the random vector $\mathbf{b}$ defined by Algorithm 2 when applied to matrix $\frac{1}{\sqrt{\gamma}}\bar{\mathbf{A}}$, and let $\mathcal{E}$ be the event that the matrix $\mathbf{Q}\mathbf{U}_{*,\mathbf{b}}$ achieves nearly-uniform marginals; i.e., $\|\mathbf{e}_i^\top\mathbf{Q}\mathbf{U}_{*,\mathbf{b}}\|^2 \le C(\tilde{\ell} + \log(m/\delta))/m$ for all i, where recall that $\mathbf{U}_{*,\mathbf{b}}$ is an $m \times \tilde{\ell}$ matrix. For any fixed $\mathbf{b}$, this event holds with probability $1-\delta^2$ due to Lemma 8. In other words, we know that $\Pr(\neg\mathcal{E} \mid \mathbf{b}) \le \delta^2$. From this, it follows that:

$$\mathbb{E}\big[\Pr(\neg\mathcal{E} \mid \mathbf{Q})\big] = \Pr(\neg\mathcal{E}) = \mathbb{E}\big[\Pr(\neg\mathcal{E} \mid \mathbf{b})\big] \le \delta^2.$$

Thus, letting $\delta_{\mathbf{Q}} = \Pr(\neg\mathcal{E} \mid \mathbf{Q})$, via Markov's inequality we have that:

$$\Pr\big(\delta_{\mathbf{Q}} \ge \delta\big) \le \frac{\mathbb{E}[\delta_{\mathbf{Q}}]}{\delta} = \frac{\delta^2}{\delta} = \delta.$$

So, for a random $\mathbf{Q}$, there is an event $\mathcal{A}$ such that $\Pr(\mathcal{A}) \ge 1-\delta$, and conditioned on $\mathcal{A}$, we achieve nearly-uniform row norms for a random $\mathbf{b}$ with probability $1-\delta$. From now on, we will condition on event $\mathcal{A}$. Next, note that a standard Chernoff bound on the sum of Bernoulli random variables implies that $\tilde{\ell} = \sum_i b_i \le C(\ell + \sqrt{\ell\log(1/\delta)})$ with probability $1-\delta$. Thus, for the rest of the analysis, we can assume that $\tilde{\ell} \le C\ell$.

Suppose that for the randomly sampled $\mathbf{b}$ the matrix $\tilde{\mathbf{U}}_{*,\mathbf{b}} = \mathbf{Q}\mathbf{U}_{*,\mathbf{b}}$ has $\tilde{\ell} \le C\ell$ columns and nearly-uniform marginals, i.e., $\|\tilde{\mathbf{u}}_i\|^2 \le C(\ell + \log(m/\delta))/m$, where $\tilde{\mathbf{u}}_i$ is the transposed

ith row of $\tilde{\mathbf{U}}_{*,\mathbf{b}}$. Next, we show that even after expanding to matrix $\bar{\mathbf{A}}$, the corresponding matrix of left singular vectors $\bar{\mathbf{S}}\tilde{\mathbf{U}}_{*,\mathbf{b}}$ also has nearly-uniform marginals with high probability. Note that in the case of $s = 1$ non-zeros per row, which corresponds to simply sub-sampling from the matrix $\mathbf{QA}$, the expansion is trivial, and so this property is immediately implied by $\tilde{\mathbf{U}}_{*,\mathbf{b}}$ having nearly-uniform rows. However, to work well with the second-moment projection analysis, we need to show the result when $s \geq C\log^4(n/\delta)$. In this case, we can only show the following somewhat weaker guarantee over all rows of $\bar{\mathbf{S}}\tilde{\mathbf{U}}_{*,\mathbf{b}}$. Let $\frac{1}{\sqrt{N}}\bar{\mathbf{s}}$ denote a row of $\bar{\mathbf{S}}$, with $\bar{\mathbf{s}} = \sqrt{\frac{m}{s}}\sum_{i=1}^{s} r_i\mathbf{e}_{I_i}^\top$. Again using $\tilde{\mathbf{u}}_i$ to denote the rows of $\tilde{\mathbf{U}}_{*,\mathbf{b}}$, we have:

$$\|\bar{\mathbf{s}}^\top\tilde{\mathbf{U}}_{*,\mathbf{b}}\| \leq \sqrt{\frac{m}{s}}\sum_{i=1}^{s}\|\tilde{\mathbf{u}}_{I_i}\| \leq \sqrt{\frac{m}{s}}\; sC\sqrt{\frac{\ell+\log(m/\delta)}{m}} \leq C\sqrt{s(\ell+\log(m/\delta))}.$$

This leads to an upper bound of $C\frac{s\ell}{N}$ for the leverage scores of $\bar{\mathbf{S}}\tilde{\mathbf{U}}_{*,\mathbf{b}}$, which is sub-optimal by a factor of s. However, fortunately we can show a sharper guarantee for a vast majority of the leverage scores. Let $\frac{1}{\sqrt{N}}\bar{\mathbf{s}}$ now be a *randomly* sampled row of $\bar{\mathbf{S}}$. Using the Gaussian universality result from Lemma 9 (specifically, the upper-bound on the largest singular value with the sketching matrix of size $1 \times m$) for $\bar{\mathbf{s}}^\top\tilde{\mathbf{U}}_{*,\mathbf{b}}$, assuming $\bar{\mathbf{s}}$ has at least $C\log^4(n/\delta)$ non-zeros, we have:

$$\Pr\left(\|\bar{\mathbf{s}}^\top\tilde{\mathbf{U}}_{*,\mathbf{b}}\| \geq C\sqrt{\ell}\right) \leq \delta, \tag{3}$$

where the probability is taken with respect to the randomness of $\bar{\mathbf{s}}$ only. We can use the above guarantees to design a probability distribution q over the row indices $\{1, ..., N\}$ of $\bar{\mathbf{S}}\tilde{\mathbf{U}}_{*,\mathbf{b}}$, such that it is nearly proportional to the row norms squared, as required by Algorithm 3.

Consider partitioning the row indices of $\bar{\mathbf{A}}$ into the set $\mathcal{R}$ of indices for which we can establish a sharp bound $\|\bar{\mathbf{s}}_i^\top\tilde{\mathbf{U}}_{*,\mathbf{b}}\| \leq C\sqrt{\ell}$, and its complement $\bar{\mathcal{R}}$, for which we merely have $\|\bar{\mathbf{s}}_i^\top\tilde{\mathbf{U}}_{*,\mathbf{b}}\| \leq C\sqrt{s(\ell+\log(m/\delta))}$, where $\bar{\mathbf{s}}_i$ denotes the transposed ith row of $\bar{\mathbf{S}}$. Formally, we define:

$$\mathcal{R} = \Big\{ i \in \{1, ..., N\} \;:\; \|\bar{\mathbf{s}}_i^\top\tilde{\mathbf{U}}_{*,\mathbf{b}}\|^2 \leq C\ell \Big\}.$$

Now, we construct the probability distribution q for Algorithm 3 as follows:

$$q_i = \begin{cases} (1-\epsilon)\frac{1}{N} & \text{if } i \in \mathcal{R}, \\ \frac{1}{\ell N}\|\bar{\mathbf{s}}_i^\top\tilde{\mathbf{U}}_{*,\mathbf{b}}\|^2 & \text{otherwise}, \end{cases}$$

where ϵ is chosen so that $\sum_i q_i = 1$. Using (3), we know that $\bar{\mathcal{R}}$ includes no more than a δ fraction of all indices, i.e., $|\bar{\mathcal{R}}|/N \leq \delta$, so we can solve for ϵ:

$$1 = \sum_i q_i = (1-\epsilon)\sum_{i\in R}\frac{1}{N} + \sum_{i\notin R}\frac{1}{\ell N}\|\bar{\mathbf{s}}_i^\top\tilde{\mathbf{U}}_{*,\mathbf{b}}\|^2 \leq (1-\epsilon) + Cs\delta,$$

so we can use $\epsilon \leq Cs\delta$. It is easy to see that as long as $\ell \geq \log(m/\delta)$, distribution q is a constant factor approximation of leverage score sampling for $\bar{\mathbf{S}}\tilde{\mathbf{U}}_{*,\mathbf{b}}$, so that it can be used in Algorithm 3 with $R = O(1)$ in Lemma 17. However, since we cannot actually compute q, we still have to show that q can be coupled with the exact uniform sampling distribution $q' = (\frac{1}{N}, ..., \frac{1}{N})$ which represents LESS-uniform. We do this by bounding the total variation distance $d_{\mathrm{tv}}(q, q')$ between q and q', as follows:

$$\begin{aligned}
d_{\mathrm{tv}}(q, q') &= \frac{1}{2}\sum_i |q(i) - q'(i)| \\
&\le \sum_{i\in\mathcal{R}} |q(i) - q'(i)| + \sum_{i\in\bar{\mathcal{R}}} |q(i) - q'(i)| \\
&\le \sum_{i\in\mathcal{R}} \frac{\epsilon}{N} + \sum_{i\in\bar{\mathcal{R}}} \Big(\frac{1}{N} + \frac{Cs\ell}{lN}\Big) \\
&\le \epsilon + Cs\delta \le 2Cs\delta.
\end{aligned}$$

So, by Lemma 19, we can couple approximate leverage score sampling q with uniform sampling q' without observing any difference for $O(\ell \log \ell)$ steps of Algorithm 3, with probability $1 - O(s\delta\ell \log \ell)$. This is sufficient since we know from Lemma 17 that the algorithm will terminate after $O(\ell \log \ell)$ steps. Note that adjusting the constants in the statement of the result we can replace the failure probability $O(s\delta\ell \log \ell)$ with δ.

To summarize, we have shown that, conditioned on the event $\mathcal{A}$ that only depends on $\mathbf{Q}$, running Algorithms 2 and 3 as described above on input $\frac{1}{\sqrt{\lambda}}\bar{\mathbf{A}} = \frac{1}{\sqrt{\lambda}}\bar{\mathbf{S}}\mathbf{Q}\mathbf{A}$ will produce a sample from $\mathcal{S}_{\mathrm{DPP}} \sim \mathrm{DPP}(\frac{1}{\lambda}\bar{\mathbf{A}}\bar{\mathbf{A}}^\top)$ by drawing indices at random from $\{1, ..., N\}$ according to distribution q. Let $\mathcal{S}$ denote the set of these indices, including the ones rejected in Algorithm 3. We showed that with probability $1 - \delta$, this set has size at most $|\mathcal{S}| = O(\ell \log(\ell/\delta))$, and that the distribution of these indices can be coupled with a uniform sample of indices so that they are indistinguishable with probability $1 - \delta$.

Rephrasing this, we can say that there is a coupling between the run of our DPP sampler (producing $\mathcal{S}_{\mathrm{DPP}}$) and a uniform sample $\mathcal{S}$ of size $k \le C\ell \log \ell$ such that, conditioned on an event $\mathcal{B}$ that holds with probability $1 - \delta$, we have $\mathcal{S}_{\mathrm{DPP}} \subseteq \mathcal{S}$. Now, let $\mathbf{P}_{\mathcal{S}_{\mathrm{DPP}}}$ and $\mathbf{P}_{\mathcal{S}}$ denote the corresponding projections onto the span of rows selected by $\mathcal{S}_{\mathrm{DPP}}$ and $\mathcal{S}$ respectively. Then, we have:

$$\begin{aligned}
\mathbb{E}[\mathbf{P}_{\mathcal{S}}] &\succeq \mathbb{E}[\mathbf{P}_{\mathcal{S}}\mathbb{1}_{\mathcal{B}}] \\
&\overset{(1)}{\succeq} \mathbb{E}[\mathbf{P}_{\mathcal{S}_{\mathrm{DPP}}}\mathbb{1}_{\mathcal{B}}] \\
&\overset{(2)}{\succeq} \mathbb{E}[\mathbf{P}_{\mathcal{S}_{\mathrm{DPP}}}] - \delta\mathbf{I} \\
&\overset{(3)}{=} \mathbf{A}^\top\mathbf{A}(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I})^{-1} - \delta\mathbf{I},
\end{aligned}$$

where (1) follows because $\mathcal{S}_{\mathrm{DPP}} \subseteq \mathcal{S}$ when conditioned on $\mathcal{B}$, (2) follows because $\mathcal{B}$ holds with $1 - \delta$ probability, and (3) follows from Lemma 15. Finally, note that while in the proof we started with ℓ and then selected $k \le C\ell \log \ell$, this can easily be restated as starting with $k \ge C \log m$ and selecting $\ell \ge \frac{Ck}{\log k}$, since $\frac{k}{\log k}\log(\frac{k}{\log k}) \le Ck$. ■

6. Second-Moment Projection Analysis via Gaussian Universality

We now turn to bounding the second-order term $\nu = \lambda_{\max}(\mathbb{E}[(\bar{\mathbf{P}}^{-1/2}\mathbf{P}\bar{\mathbf{P}}^{-1/2})^2])$, which appears alongside the first-order term μ in the convergence bound (1) of Sketch-and-Project with Nesterov's acceleration (Algorithm 1). We do this by combining the previous bound on the expected projection with a bound on the smallest singular value of the LESS-uniform sketch $\mathbf{SA}$. Our main result is a more general statement on a positive semidefinite ordering of a certain second matrix moment of $\mathbf{P}$.

Lemma 20 *Suppose that a rank n matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with condition number κ is transformed by the randomized Hadamard transform (RHT), i.e., $\mathbf{A} \leftarrow \mathbf{QA}$. If $\mathbf{S}$ is a $k \times m$ LESS-uniform sketching matrix with sketch size $k \geq C_1 \log(m/\delta)$, $2k \leq m$ and $s \geq C_2 \log^4(n/\delta)$ non-zeros per row, $n \geq C_3 \log(1/\delta)$, and $\delta \leq C_4/n\kappa^2$, then, conditioned on an RHT property that holds with probability $1-\delta$, matrix $\mathbf{P} = (\mathbf{SA})^\dagger\mathbf{SA}$ satisfies:*

$$\mathbb{E}\big[\mathbf{P}\mathbb{E}[\mathbf{P}]^{-1}\mathbf{P}\big] \preceq \frac{c_1 n}{\ell}\bar{\kappa}^2_{\ell:2k} \cdot \mathbb{E}[\mathbf{P}] + 2\delta\mathbf{I} \quad \textit{for} \quad \ell = \left\lceil \frac{c_2 k}{\log k}\right\rceil,$$

where $\bar{\kappa}_{\ell:q} := (\frac{1}{q-\ell}\sum_{i=\ell+1}^q \frac{\sigma_i^2}{\sigma_q^2})^{1/2}$ with $\sigma_1 \geq \sigma_2 \geq \ldots$ denoting the singular values of $\mathbf{A}$.

From this main result, we can conclude the following bound on ν.

Corollary 21 *Under the assumptions of Lemma 20,*

$$\nu \leq \frac{c_1 n}{\ell}\bar{\kappa}^2_{\ell:2k} \quad \textit{for} \quad \ell \geq \frac{c_2 k}{\log k}. \tag{4}$$

We will prove Lemma 20 and Corollary 21 together, using the following lemma which allows us to control the smallest singular value of the LESS-uniform sketch with high probability.

Lemma 22 *Suppose that we are given matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with full column rank, and let random matrix $\mathbf{Q} \in \mathbb{R}^{m\times m}$ be the randomized Hadamard transform. If the sketching matrix $\mathbf{S}$ is a $k \times m$ LESS-uniform matrix with $C_1 \log(m/\delta) \leq 2k \leq m$ and $s \geq C_2 \log^4(n/\delta)$ non-zeros per row, then*

$$\sigma^2_{\min}(\mathbf{SQA}) \geq c\Big(2k\sigma_{2k}^2 + \sum_{i>2k}\sigma_i^2\Big) \quad \textit{with probability at least } 1-\delta.$$

Remark 4 *Our argument is quite general, and in particular, can be easily adapted to sketching methods where $\mathbf{SQ}$ is replaced by a Gaussian or a sub-Gaussian matrix. To our knowledge, this is the first such characterization for the smallest singular value of a sketch.*

Proof First, we note that $\sigma^2_{\min}(\mathbf{SQA}) = \lambda_{\min}(\mathbf{SQAA}^\top\mathbf{Q}^\top\mathbf{S}^\top)$. Let $\mathbf{A} = \mathbf{UDV}^\top$ be the singular value decomposition (SVD) of $\mathbf{A}$, and let the corresponding SVD of $\mathbf{QA}$ be $\tilde{\mathbf{U}}\mathbf{DV}^\top$ where $\tilde{\mathbf{U}} = \mathbf{QU}$ and we used the fact that $\mathbf{Q}$ is an orthogonal matrix. The central part of our argument is to use the SVD of $\mathbf{A}$ to decompose the matrix $\mathbf{SQAA}^\top\mathbf{Q}^\top\mathbf{S}^\top$ into a sum of isotropic random matrices via a careful telescoping argument. This allows us to apply Gaussian universality of the smallest singular value (Lemma 9) to each component of the

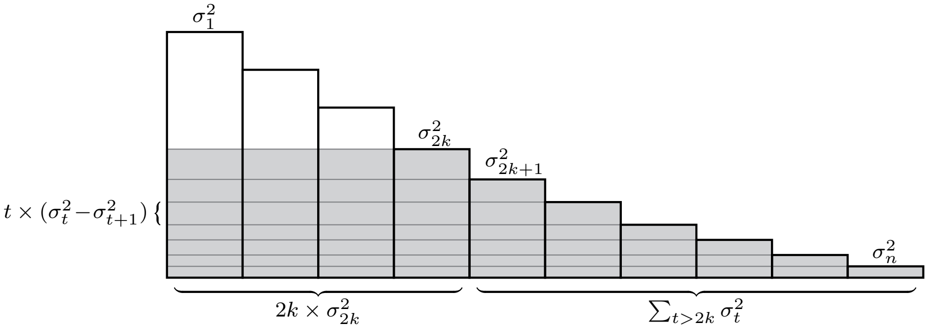

Figure 1: Illustration for the telescoping sum argument in the proof of Lemma 22. The expression in (6) can be viewed as computing the shaded area by summing over the row rectangles, whereas the expression in (7) is computing the same shaded are by summing over the columns.

sum. To that end, letting $\tilde{\mathbf{U}}_{*,t}$ denote the t-th column of $\tilde{\mathbf{U}}$, and $\tilde{\mathbf{U}}_{*,1:t}$ be the $m \times t$ matrix consisting of the first t columns of $\tilde{\mathbf{U}}$, we start by lower bounding $\mathbf{SQAA}^\top\mathbf{Q}^\top\mathbf{S}^\top$ with the following telescoping sum expression:

$$\begin{aligned}\mathbf{SQAA}^\top\mathbf{Q}^\top\mathbf{S}^\top &= \mathbf{S}\tilde{\mathbf{U}}\mathbf{D}\mathbf{V}^\top\mathbf{V}\mathbf{D}\tilde{\mathbf{U}}^\top\mathbf{S}^\top = \mathbf{S}\tilde{\mathbf{U}}\mathbf{D}^2\tilde{\mathbf{U}}^\top\mathbf{S}^\top \\ &= \sum_{t=1}^{n} \sigma_t^2 \mathbf{S}\tilde{\mathbf{U}}_{*,t}\tilde{\mathbf{U}}_{*,t}^\top\mathbf{S}^\top \\ &\succeq \sum_{t\geq 2k}^{n-1} (\sigma_t^2 - \sigma_{t+1}^2)\mathbf{S}\tilde{\mathbf{U}}_{*,1:t}\tilde{\mathbf{U}}_{*,1:t}^\top\mathbf{S}^\top + \sigma_n^2\mathbf{S}\tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top\mathbf{S}^\top \\ &\succeq \Big(\sum_{t\geq 2k}^{n-1} (\sigma_t^2 - \sigma_{t+1}^2)\sigma_{\min}^2(\mathbf{S}\tilde{\mathbf{U}}_{*,1:t}) + \sigma_n^2\sigma_{\min}^2(\mathbf{S}\tilde{\mathbf{U}})\Big)\cdot\mathbf{I}, \qquad (5)\end{aligned}$$

where, at a high level, the idea is to combine the contributions of the rank one matrices $\tilde{\mathbf{U}}_{*,t}\tilde{\mathbf{U}}_{*,t}^\top$ (corresponding to the t-th singular vector) into groups of $2k$ or more, so that all of the matrices $\mathbf{S}\tilde{\mathbf{U}}_{*,1:t}$ appearing in (5) are sufficiently wide for us to be able to show a lower bound on the smallest singular values of $\sigma_{\min}^2(\mathbf{S}\tilde{\mathbf{U}}_{*,1:t})$ via Lemma 9.

To that end, note that each matrix $\tilde{\mathbf{U}}_{*,1:t}$ has orthonormal columns, and since it can be written as $\tilde{\mathbf{U}}_{*,1:t} = \mathbf{Q}\mathbf{U}_{*,1:t}$, we can use the row uniformizing property of the randomized Hadamard transform $\mathbf{Q}$. Namely, for each value of $t \geq 2k$, we can apply Lemma 8 to obtain that with probability $1 - \delta/2n$, the row norms of $\tilde{\mathbf{U}}_{*,1:t}$ are uniformly bounded as

$$\|\tilde{\mathbf{U}}_{i,1:t}\| \leq \sqrt{t/m} + \sqrt{8\log(2mn/\delta)/m} \leq C\sqrt{t/m}.$$

This means that each matrix $\tilde{\mathbf{U}}_{*,1:t}$ fits the assumptions of our Gaussian universality result (Lemma 9), and we can use it to lower bound the smallest singular value. Specifically, the lemma implies that for any $t \geq 2k$, with probability $1 - \delta/2n$ the matrix $\mathbf{S}\tilde{\mathbf{U}}_{*,1:t} \in \mathbb{R}^{k\times t}$

satisfies

$$\sigma_{\min}(\mathbf{S}\tilde{\mathbf{U}}_{*,1:t}) \geq c_1\sqrt{t}.$$

Taking a union bound, we can ensure that this singular value lower bound holds for every t with probability $1-\delta$. Plugging these inequalities into the telescoping sum in (5), we obtain:

$$\sigma^2_{\min}(\mathbf{SQA}) \geq c_1^2\bigg(\sum_{t\geq 2k}^{n-1} t(\sigma_t^2 - \sigma_{t+1}^2) + n\sigma_n^2\bigg). \tag{6}$$

To reach the final claimed lower bound, it remains to undo the telescoping sum expression. This can be explained most simply via the diagram in Figure 1. Here, the expression in (6) can be interpreted as computing the shaded area in the figure by summing up over the row rectangles of dimension $t \times (\sigma_t^2 - \sigma_{t+1}^2)$. Instead, we can compute it by summing over the columns, i.e., the total contributions of each singular value. This breaks the expression down into the term that corresponds to the $2k \times \sigma_{2k}^2$ rectangle, plus the sum over the tail of the remaining singular values. Thus, we obtain:

$$\sigma^2_{\min}(\mathbf{SQA}) \geq c_1^2\bigg(2k\sigma_{2k}^2 + \sum_{t=2k+1}^{n} \sigma_t^2\bigg). \tag{7}$$

This concludes the proof. ■

We are now ready to prove Lemma 20; i.e., provide an upper-bound for the second-order projection expression $\mathbb{E}[\mathbf{P}\mathbb{E}[\mathbf{P}]^{-1}\mathbf{P}]$, along with Corollary 21. Recall that the random projection matrix is defined as $\mathbf{P} = (\mathbf{SA})^\dagger\mathbf{SA}$, where to simplify the notation we assume that the matrix $\mathbf{A}$ has already been preprocessed with the RHT matrix $\mathbf{Q}$. **Proof of Lemma 20 and Corollary 21** We start by bounding the term $\mathbb{E}[\mathbf{P}]^{-1}$ inside the expression. For this, we can rely on the first-order analysis from Lemma 12, which states that:

$$\mathbb{E}[\mathbf{P}] \succeq \mathbf{A}^\top\mathbf{A}(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I})^{-1} - \delta\mathbf{I},$$

where $\lambda = \frac{1}{\ell}\sum_{i>\ell}\sigma_i^2$ and $\ell = \lceil\frac{ck}{\log k}\rceil$. Now, since by the assumption in the lemma we have $\delta \leq 1/2n\kappa^2 \leq \lambda_{\min}(\mathbf{A}^\top\mathbf{A}(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})^{-1})/2$, we have

$$\mathbb{E}[\mathbf{P}]^{-1} \preceq 2(\mathbf{A}^\top\mathbf{A})^{-1}(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I}) = 2\mathbf{I} + 2\lambda(\mathbf{A}^\top\mathbf{A})^{-1}.$$

Applying this bound to our second-order expression, we obtain:

$$\begin{aligned}\mathbb{E}\big[\mathbf{P}\mathbb{E}[\mathbf{P}]^{-1}\mathbf{P}\big] &\preceq 2\mathbb{E}\Big[\mathbf{P}\big(\mathbf{I} + \lambda(\mathbf{A}^\top\mathbf{A})^{-1}\big)\mathbf{P}\Big] \\ &= 2\mathbb{E}[\mathbf{P}] + 2\lambda\mathbb{E}[\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}].\end{aligned}$$

Thus, it remains to bound $\mathbb{E}[\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}]$. First, we will use Lemma 22 to define a high-probability event that bounds the spectral norm of the matrix $\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}$. To do that, define $\mathbf{U} = \mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1/2}$, which is a matrix with orthonormal columns (recall that we assumed full column rank of $\mathbf{A}$). By expanding the Moore-Penrose pseudoinverse we can write $\mathbf{P} = (\mathbf{SA})^\dagger \mathbf{SA} = \mathbf{A}^\top\mathbf{S}^\top(\mathbf{SAA}^\top\mathbf{S}^\top)^{-1}\mathbf{SA}$, which gives us:

$$\begin{aligned}\|\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}\| &= \|(\mathbf{A}^\top\mathbf{A})^{-1/2}\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1/2}\| \\ &= \|\mathbf{U}^\top\mathbf{S}^\top(\mathbf{SAA}^\top\mathbf{S}^\top)^{-1}\mathbf{SU}\| \\ &\le \|\mathbf{SU}\|^2\|(\mathbf{SAA}^\top\mathbf{S}^\top)^{-1}\| = \frac{\sigma^2_{\max}(\mathbf{SU})}{\sigma^2_{\min}(\mathbf{SA})}.\end{aligned}$$

This allows us to leverage Lemma 22, ensuring that $\sigma^2_{\min}(\mathbf{SA})$ is lower bounded by $c(2k\sigma^2_{2k} + \sum_{i>2k}\sigma_i^2)$ with probability $1-\delta/2$. Moreover, since $\mathbf{U}$ is a matrix with orthonormal columns that was preprocessed by an RHT, we can use Lemma 9 to control $\sigma^2_{\max}(\mathbf{SU})$, obtaining that with probability $1-\delta/2$:

$$\sigma_{\max}(\mathbf{SU}) \le c_2\sqrt{n}. \tag{8}$$

Putting these bounds together, we can define the following event which, for an appropriate constant $C > 0$, holds with probability $1-\delta$:

$$\mathcal{E} := \Big\{\|\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}\| \le \frac{Cn}{2k\sigma^2_{2k} + \sum_{i>2k}\sigma_i^2}\Big\}.$$

Unfortunately, the above bound on the spectral norm of $\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}$ is not sharp enough by itself when used to bound the expectation of this matrix. However, we are able get a sharper bound on $\mathbb{E}[\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}]$ in the positive semidefinite ordering. First, using law of total expectation,

$$\begin{aligned}\mathbb{E}[\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}] &= \mathbb{E}[\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P} \mid \mathcal{E}]\Pr(\mathcal{E}) + \mathbb{E}[\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P} \mid \neg\mathcal{E}]\Pr(\neg\mathcal{E}) \\ &\preceq \mathbb{E}[\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}\cdot 1_{\mathcal{E}}] + \delta\sigma_{\min}^{-2}\mathbb{E}[\mathbf{P} \mid \neg\mathcal{E}],\end{aligned}$$

where $1_{\mathcal{E}}$ is the characteristic variable of $\mathcal{E}$. The second term is small thanks to δ, while for the first term we can use the spectral norm bound, but in a careful way. Recalling that $\mathbf{Z}^\top\mathbf{AZ} \preceq \mathbf{Z}^\top\mathbf{BZ}$ if $\mathbf{A} \preceq \mathbf{B}$, and using that $\mathbf{P} = \mathbf{P}^2$ for a projection matrix:

$$\begin{aligned}\mathbb{E}[\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}\cdot 1_{\mathcal{E}}] &= \mathbb{E}[\mathbf{P}\cdot\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}\cdot\mathbf{P}\cdot 1_{\mathcal{E}}] \\ &\preceq \frac{Cn}{2k\sigma^2_{2k} + \sum_{i>2k}\sigma_i^2}\mathbb{E}[\mathbf{P}\cdot 1_{\mathcal{E}}].\end{aligned}$$

Putting those together, since $\mathbb{E}[\mathbf{P}\cdot 1_{\mathcal{E}}] \preceq \mathbb{E}[\mathbf{P}]$ and $\mathbf{P} \preceq \mathbf{I}$:

$$\mathbb{E}[\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}] \preceq \frac{Cn}{2k\sigma^2_{2k} + \sum_{i>2k}\sigma_i^2}\mathbb{E}[\mathbf{P}] + \delta\sigma_{\min}^{-2}\mathbf{I}.$$

Now, we are ready to return to the second-order expression $\mathbb{E}[\mathbf{P}\mathbb{E}[\mathbf{P}]^{-1}\mathbf{P}]$.

$$\begin{aligned}\mathbb{E}\big[\mathbf{P}\mathbb{E}[\mathbf{P}]^{-1}\mathbf{P}\big] &\preceq 2\mathbb{E}[\mathbf{P}] + 2\lambda\mathbb{E}[\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{P}] \\ &\preceq 2\mathbb{E}[\mathbf{P}] + \frac{2C\lambda n}{2k\sigma^2_{2k} + \sum_{i>2k}\sigma_i^2}\mathbb{E}[\mathbf{P}] + 2\lambda\delta\sigma_{\min}^{-2}\mathbf{I} \\ &= 2\Big(1 + C\frac{n}{\ell}\gamma\Big)\mathbb{E}[\mathbf{P}] + 2\lambda\delta\sigma_{\min}^{-2}\mathbf{I}, \quad \text{where} \quad \gamma = \frac{\sum_{i>\ell}\sigma_i^2}{2k\sigma^2_{2k} + \sum_{i>2k}\sigma_i^2}.\end{aligned}$$

Finally, to conclude the proof, note that $\lambda \geq \sigma_{\min}^2$ and that

$$\begin{aligned}\gamma &= \frac{\sum_{i>\ell}\sigma_i^2}{2k\sigma_{2k}^2 + \sum_{i>2k}\sigma_i^2}\\ &= \frac{\sum_{i>\ell}^{2k}\sigma_i^2}{2k\sigma_{2k}^2 + \sum_{i>2k}\sigma_i^2} + \frac{\sum_{i>2k}\sigma_i^2}{2k\sigma_{2k}^2 + \sum_{i>2k}\sigma_i^2}\\ &\leq \frac{1}{2k}\sum_{i>\ell}^{2k}\frac{\sigma_i^2}{\sigma_{2k}^2} + 1 \leq \bar{\kappa}_{\ell:2k}^2 + 1.\end{aligned}$$

To bound ν, we can simply multiply $\mathbb{E}[\mathbf{P}\mathbb{E}[\mathbf{P}]^{-1}\mathbf{P}]$ by $\mathbb{E}[\mathbf{P}]^{-1/2}$ from both sides, obtaining $\mathbb{E}[(\mathbb{E}[\mathbf{P}]^{-1/2}\mathbf{P}\mathbb{E}[\mathbf{P}]^{-1/2})^2]$, and then take the spectral norm, noting that

$$\delta\|\mathbb{E}[\mathbf{P}]^{-1}\| = \frac{\delta}{\mu} \leq \frac{cn}{\ell}\frac{\bar{\kappa}_\ell^2}{n\kappa^2} \leq \frac{cn}{\ell},$$

and so the contribution of the δ can be absorbed into the constant. ■

7. Convergence Analysis with Inexact Projections

Step 6 of Algorithm 1 requires finding $\mathbf{u}_t$ that solves the following linear system:

$$(\mathbf{S}\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}^\top)\mathbf{u}_t = \tilde{\mathbf{b}} \quad \text{where} \quad \tilde{\mathbf{b}} = \mathbf{S}\mathbf{A}\mathbf{y}_t - \mathbf{S}\mathbf{b}. \tag{9}$$

Then, in Step 7, the algorithm computes $\mathbf{w}_t = \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}^\top\mathbf{u}_t$. The cost of solving (9) is the main cost involved in each iteration of sketch-and-project. Recall that in our main results, we consider two different choices of $\mathbf{B}$ depending on whether $\mathbf{A}$ is positive definite or not.

Positive definite case. In the case where matrix $\mathbf{A}$ is positive definite, we choose $\mathbf{B} = \mathbf{A}$, which means that the linear system we must solve becomes $(\mathbf{S}\mathbf{A}\mathbf{S}^\top)\mathbf{u}_t = \tilde{\mathbf{b}}$. The matrix $\mathbf{S}\mathbf{A}\mathbf{S}^\top$ can be computed in time $\tilde{O}(nk)$ and inverted in time $O(k^\omega)$, so we can solve the projection step (9) exactly in time $\tilde{O}(nk + k^\omega)$.

General case. On the other hand, in the case of solving a general linear system, where we choose $\mathbf{B} = \mathbf{I}$, the resulting linear system is:

$$(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top)\mathbf{u}_t = \tilde{\mathbf{b}}, \quad \text{for} \quad \tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}.$$

Here, the matrix $\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}$ can be precomputed in $\tilde{O}(nk)$ time, however solving the linear system requires computing the matrix $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top$, which would take $O(nk^{\omega-1})$ to do exactly. To avoid that, we instead solve this projection step approximately using Preconditioned Conjugate Gradient (PCG), which will allow us to recover the same $\tilde{O}(nk + k^\omega)$ time as in the positive definite case.

Naturally, the quality of the approximation in the projection step influences the convergence rate of Algorithm 1. Our next Lemma 23 shows that as soon as $\mathbf{w}_t$ is estimated with small enough error, Algorithm 1 converges in expectation with convergence rate determined by μ and ν that essentially matches the rate achieved with an exact projection step. For the sake of simplicity, we present this result only for the case of $\mathbf{B} = \mathbf{I}$.

Lemma 23 (Algorithm 1 with inexact projections) *Suppose that* $\mathbf{B} = \mathbf{I}$, *and steps 6, 7 of Algorithm 1 result in an estimate* $\hat{\mathbf{w}}_t \in range(\mathbf{A}^\top)$ *of* $\mathbf{w}_t = \tilde{\mathbf{A}}^\top(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top)^{-1}\tilde{\mathbf{b}}$ *such that*

$$\|\hat{\mathbf{w}}_t - \mathbf{w}_t\| \leq \epsilon\|\mathbf{w}_t\| \quad \text{with} \quad \epsilon \leq \frac{\mu}{\sqrt{8\|\mathbb{E}[\mathbf{P}]\|(\mu\nu+1)}}.$$

Then, with $\beta = 1 - \frac{3}{4}\sqrt{\frac{\mu}{\nu}}$, $\gamma = \frac{1}{\sqrt{\mu\nu}}$, $\alpha = \frac{1}{1+\gamma\nu}$, *we have that:*

$$\mathbb{E}\big[\Delta_t\big] \leq \left(1 - \frac{1}{2}\sqrt{\frac{\mu}{\nu}}\right)^t \Delta_0,$$

where the error function is defined as $\Delta_t := \|\mathbf{v}_t - \mathbf{x}^*\|^2_{\mathbb{E}[\mathbf{P}]^\dagger} + \frac{1}{\mu}\|\mathbf{x}_t - \mathbf{x}^*\|^2$.

Remark 5 *Exact knowledge of* μ *and* ν *is not required: Given* $\tilde{\mu}$ *and* $\tilde{\nu}$ *such that* $\mu \geq \tilde{\mu}$ *and* $\nu \leq \tilde{\nu}$, *the hyperparameters* α, β, *and* γ *can be set using* $\tilde{\mu}$ *and* $\tilde{\nu}$ *in place of* μ *and* ν, *and an identical convergence analysis recovers the rate of* $\mathbb{E}[\Delta_t] \leq \big(1 - \frac{1}{2}\sqrt{\tilde{\mu}/\tilde{\nu}}\big)^t\Delta_0$ *under the inexact projection condition with* $\epsilon = \frac{\tilde{\mu}}{\sqrt{8(\tilde{\mu}\tilde{\nu}+1)}}$, *where we used that* $\|\mathbb{E}[\mathbf{P}]\| \leq 1$.

Remark 6 *When Algorithm 1 uses exact projections (steps 6 and 7), then Gower et al. (2018) showed a convergence bound which can be stated for a general positive definite* $\mathbf{B}$. *Redefining*

$$\mathbf{P} := \mathbf{B}^{-1/2}\tilde{\mathbf{A}}^\top(\tilde{\mathbf{A}}\mathbf{B}^{-1}\tilde{\mathbf{A}}^\top)^{-1}\tilde{\mathbf{A}}\mathbf{B}^{-1/2} \quad \text{and}$$
$$\Delta_t := \|\mathbf{v}_t - \mathbf{x}^*\|^2_{\mathbf{B}^{1/2}\mathbb{E}[\mathbf{P}]^\dagger\mathbf{B}^{1/2}} + \frac{1}{\mu}\|\mathbf{x}_k - \mathbf{x}^*\|^2_{\mathbf{B}},$$

they showed the following convergence bound for the exact implementation of Algorithm 1:

$$\mathbb{E}\big[\Delta_t\big] \leq \left(1 - \sqrt{\frac{\mu}{\nu}}\right)^t \Delta_0,$$

Our Lemma 23 gets the same rate up to a constant factor $1/2$ *without requiring exact projection steps (for the case of* $\mathbf{B} = \mathbf{I}$*). We note that the constant* $1/2$ *can be improved to any constant* $c < 1$ *by tuning the other parameters of Lemma 23, along the same lines.*

Before we proceed with the proof of Lemma 23, we now explain how we use an iterative solver to compute $\mathbf{u}_t$ and $\mathbf{w}_t$ approximately when $\mathbf{B} = \mathbf{I}$. Taking advantage of the fact that the matrix $\tilde{\mathbf{A}}$ is very wide, we can use sketching to construct an effective preconditioner for the linear system (9). Following Dereziński and Yang (2024), our preconditioner is computed as follows:

1. Compute $\mathbf{\Phi}\tilde{\mathbf{A}}^\top$, where $\mathbf{\Phi}$ is a $\phi \times n$ sketching matrix;
2. Construct $\mathbf{M} = \mathbf{V}\mathbf{\Sigma}^{-1}$, where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is the compact SVD of $\mathbf{\Phi}\tilde{\mathbf{A}}^\top$.

Here, we refer to a standard sketch-and-precondition guarantee to show that PCG with $\mathbf{M}$ as a preconditioner will solve the above linear system in time $\tilde{O}(nk + k^\omega)$.

Lemma 24 (Based on Lemma 7.1, Dereziński and Yang (2024)) *Given $\tilde{\mathbf{A}} \in \mathbb{R}^{k\times n}$, $\tilde{\mathbf{b}} \in \mathbb{R}^k$, let $\mathbf{w}^* = \tilde{\mathbf{A}}^\top(\tilde{\mathbf{A}}\tilde{\mathbf{A}})^\top\tilde{\mathbf{b}}$. Given $\delta < 1/2$, there is a sparse sketching matrix matrix $\mathbf{\Phi} \in \mathbb{R}^{\phi\times n}$ that satisfies $\phi = O(k + \log(1/\delta))$ such that we can compute $\mathbf{M}$ in time $O(nk\log(k/\delta) + k^\omega)$ that with probability $1-\delta$ satisfies $\kappa(\tilde{\mathbf{A}}^\top\mathbf{M}) = O(1)$, in which case, PCG preconditioned with $\mathbf{M}$ will solve the system $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top\mathbf{u} = \tilde{\mathbf{b}}$ in time $O(nk\log 1/\epsilon)$, returning $\hat{\mathbf{u}}$ such that*

$$\|\hat{\mathbf{w}} - \mathbf{w}^*\| \le \epsilon\|\mathbf{w}^*\|,$$

where $\hat{\mathbf{w}} = \tilde{\mathbf{A}}^\top\mathbf{u}$. Also, with probability 1 and independently of $\mathbf{M}$, $\|\hat{\mathbf{w}} - \mathbf{w}^*\| \le 4\|\mathbf{w}^*\|$.

We now return to the convergence analysis of accelerated sketch-and-project with inexact projections. Recall that, for the sake of simplicity, we show this with matrix $\mathbf{B} = \mathbf{I}$, which is used in Theorem 10. **Proof of Lemma 23** Let $\mathbf{e}_t := \hat{\mathbf{w}}_t - \mathbf{w}_t$, so that by assumption $\|\mathbf{e}_t\| \le \epsilon\|\mathbf{w}_t\|$. Note that by definition,

$$\mathbf{w}_t = \tilde{\mathbf{A}}^\top\mathbf{u} = \mathbf{A}^\top\mathbf{S}^\top(\mathbf{S}\mathbf{A}\mathbf{A}^\top\mathbf{S}^\top)^\dagger\mathbf{S}(\mathbf{A}\mathbf{y}_t - \mathbf{b}) =: \mathbf{P}(\mathbf{y}_t - \mathbf{x}^*).$$

So, conditioned on the tth iteration, since $\mathbf{P}$ is idempotent,

$$\mathbb{E}\big[\|\mathbf{e}_t\|^2\big] \le \epsilon^2\mathbb{E}\big[\|\mathbf{w}_t\|^2\big] \le \epsilon^2\mathbb{E}\big[(\mathbf{y}_t - \mathbf{x}^*)^\top\mathbf{P}(\mathbf{y}_t - \mathbf{x}^*)\big] \le \epsilon^2 a\|\mathbf{y}_t - \mathbf{x}^*\|^2, \tag{10}$$

where $a := \|\mathbb{E}[\mathbf{P}]\|$ and thus we have with $\mu = \lambda_{\min}(\mathbb{E}[\mathbf{P}])$

$$\mathbb{E}\big[\|\mathbf{e}_t\|^2_{\mathbb{E}[\mathbf{P}]^\dagger}\big] = \mathbb{E}\big[\mathbf{e}_t^\top\mathbb{E}[\mathbf{P}]^\dagger\mathbf{e}_t\big] \le \|\mathbb{E}[\mathbf{P}]^\dagger\| \cdot \mathbb{E}\big[\|\mathbf{e}_t\|^2\big] \le \frac{a\epsilon^2}{\mu}\|\mathbf{y}_t - \mathbf{x}^*\|^2. \tag{11}$$

Then, for $r_t = \|\mathbf{v}_t - \mathbf{x}^*\|_{\mathbb{E}[\mathbf{P}]^\dagger}$, observe that

$$\begin{aligned}
\mathbb{E}[r_{t+1}^2] &= \|\mathbf{v}_{t+1} - \mathbf{x}^*\|^2_{\mathbb{E}[\mathbf{P}]^\dagger} \\
&= \mathbb{E}\big[\|\beta\mathbf{v}_t + (1-\beta)\mathbf{y}_t - \mathbf{x}^* - \gamma\mathbf{P}(\mathbf{y}_t - \mathbf{x}^*) - \gamma\mathbf{e}_t\|^2_{\mathbb{E}[\mathbf{P}]^\dagger}\big] \\
&\le \mathbb{E}\Big[\Big(\|\beta\mathbf{v}_t + (1-\beta)\mathbf{y}_t - \mathbf{x}^* - \gamma\mathbf{P}(\mathbf{y}_t - \mathbf{x}^*)\|_{\mathbb{E}[\mathbf{P}]^\dagger} + \|\gamma\mathbf{e}_t\|_{\mathbb{E}[\mathbf{P}]^\dagger}\Big)^2\Big] \\
&\le (1+\eta)\mathbb{E}\big[\|\beta\mathbf{v}_t + (1-\beta)\mathbf{y}_t - \mathbf{x}^* - \gamma\mathbf{P}(\mathbf{y}_t - \mathbf{x}^*)\|^2_{\mathbb{E}[\mathbf{P}]^\dagger}\big] + \left(1 + \frac{1}{\eta}\right)\mathbb{E}\big[\|\gamma\mathbf{e}_t\|^2_{\mathbb{E}[\mathbf{P}]^\dagger}\big],
\end{aligned}$$

where in the last inequality we have used the fact that for any $w, z \in \mathbb{R}$ and for any $\eta > 0$ it holds that $2wz \le \eta w^2 + \frac{1}{\eta}z^2$ (we will choose η later). Then, using (11) and expanding the square, we get

$$\mathbb{E}[r_{t+1}^2] \le (1+\eta)I + (1+\eta)\gamma^2 II - 2(1+\eta)\gamma III + \left(1 + \frac{1}{\eta}\right)\gamma^2\frac{a\epsilon^2}{\mu}\|\mathbf{y}_t - \mathbf{x}^*\|^2, \tag{12}$$

$$\begin{aligned}
\text{where} \qquad I &= \|\beta\mathbf{v}_t + (1-\beta)\mathbf{y}_t - \mathbf{x}^*\|^2_{\mathbb{E}[\mathbf{P}]^\dagger}, \\
II &= \mathbb{E}\big[\|\mathbf{P}(\mathbf{y}_t - \mathbf{x}^*)\|^2_{\mathbb{E}[\mathbf{P}]^\dagger}\big], \\
III &= \mathbb{E}\Big[\Big\langle\beta(\mathbf{v}_t - \mathbf{x}^*) + (1-\beta)(\mathbf{y}_t - \mathbf{x}^*), \mathbb{E}[\mathbf{P}]^\dagger\mathbf{P}(\mathbf{y}_t - \mathbf{x}^*)\Big\rangle\Big].
\end{aligned}$$

The terms I, II, III were estimated in Gower et al. (2018) to get

$$\begin{aligned}
&I \le \beta r_t^2 + (1-\beta)\|\mathbf{y}_t - \mathbf{x}^*\|^2_{\mathbb{E}[\mathbf{P}]^\dagger},\\
&\mathbb{E}[II|\mathbf{y}_t] \le \nu\|\mathbf{y}_t - \mathbf{x}^*\|^2_{\mathbb{E}[\mathbf{P}]^\dagger}, \qquad (13)\\
&\mathbb{E}[III|\mathbf{y}_t, \mathbf{v}_t, \mathbf{x}_t] = \|\mathbf{y}_t - \mathbf{x}^*\|^2 - \beta\frac{1-\alpha}{2\alpha}(\|\mathbf{x}_t - \mathbf{x}^*\|^2 - \|\mathbf{y}_t - \mathbf{x}_t\|^2 - \|\mathbf{y}_t - \mathbf{x}^*\|^2).
\end{aligned}$$

To estimate further, we note

$$\begin{aligned}
\mathbb{E}[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2|\mathbf{y}_t] &= \mathbb{E}[\|(\mathbf{I} - \mathbf{P})(\mathbf{y}_t - \mathbf{x}^*) - \mathbf{e}_t\|^2|\mathbf{y}_t]\\
&\le (1+\eta)\|\mathbf{y}_t - \mathbf{x}^*\|^2 - (1+\eta)\|\mathbf{y}_t - \mathbf{x}^*\|^2_{\mathbb{E}[\mathbf{P}]} + \left(1 + \frac{1}{\eta}\right)\mathbb{E}\big[\|\mathbf{e}_t\|^2\big],
\end{aligned}$$

and thus

$$\|\mathbf{y}_t - \mathbf{x}^*\|^2_{\mathbb{E}[\mathbf{P}]} \le \|\mathbf{y}_t - \mathbf{x}^*\|^2 - \frac{1}{1+\eta}\mathbb{E}[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2|\mathbf{y}_t] + \frac{1+1/\eta}{1+\eta}\mathbb{E}\big[\|\mathbf{e}_t\|^2\big]. \qquad (14)$$

Combining estimates (13) and (14) back in (12), and using (10), we conclude

$$\begin{aligned}
\mathbb{E}[r_{t+1}^2|\mathbf{y}_t, \mathbf{v}_t, \mathbf{x}_t] \le{}& (1+\eta)\beta r_t^2 + (1+\eta)\frac{(1-\beta)}{\mu}\|\mathbf{y}_t - \mathbf{x}^*\|^2 \qquad (15)\\
&+ (1+\eta)\gamma^2\nu\left(\|\mathbf{y}_t - \mathbf{x}^*\|^2 - \frac{1}{1+\eta}\mathbb{E}[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2|\mathbf{y}_t] + \frac{1+1/\eta}{1+\eta}\epsilon^2 a\|\mathbf{y}_t - \mathbf{x}^*\|^2\right)\\
&+ 2(1+\eta)\gamma\left(-\|\mathbf{y}_t - \mathbf{x}^*\|^2 + \beta\frac{1-\alpha}{2\alpha}(\|\mathbf{x}_t - \mathbf{x}^*\|^2 - \|\mathbf{y}_t - \mathbf{x}^*\|^2)\right)\\
&+ \left(\frac{1+\eta}{\eta}\right)\frac{\gamma^2 a\epsilon^2}{\mu}\|\mathbf{y}_t - \mathbf{x}^*\|^2.
\end{aligned}$$

Or equivalently,

$$\begin{aligned}
&\mathbb{E}[r_{t+1}^2 + \gamma^2\nu\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2|\mathbf{y}_t, \mathbf{v}_t, \mathbf{x}_t]\\
&\qquad\le (1+\eta)\beta r_t^2 + (1+\eta)\beta\underbrace{\gamma\frac{1-\alpha}{\alpha}}_{P_1}\|\mathbf{x}_t - \mathbf{x}^*\|^2 \qquad (16)\\
&\qquad+ (1+\eta)\underbrace{\left(\frac{(1-\beta)}{\mu} + \gamma^2\nu - 2\gamma - \beta\gamma\frac{1-\alpha}{\alpha} + \frac{\gamma^2 a\epsilon^2}{\eta}(\nu + \frac{1}{\mu})\right)}_{P_2}\|\mathbf{y}_t - \mathbf{x}^*\|^2.
\end{aligned}$$

Note that with $\alpha = (1+\gamma\nu)^{-1}$ and $\gamma = 1/\sqrt{\nu\mu}$ we have $P_1 = \gamma^2\nu = 1/\mu$.

It remains to choose ϵ and η so that $P_2 \le 0$ and $(1+\eta)\beta \le 1 - \frac{1}{2}\sqrt{\frac{\mu}{\nu}}$. With $\beta = 1 - \frac{3}{4}\sqrt{\frac{\mu}{\nu}}$, the second inequality is achieved by taking $\eta = \frac{1}{4}\sqrt{\frac{\mu}{\nu}}$. The same parameters achieve $P_2 \le 0$ as soon as $\epsilon \le \frac{\mu}{\sqrt{8a(\mu\nu+1)}}$. Then, we have

$$\mathbb{E}\left[r_{t+1}^2 + \frac{1}{\mu}\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 \mid \mathbf{y}_t, \mathbf{v}_t, \mathbf{x}_t\right] \le \left(1 - \frac{1}{2}\sqrt{\frac{\mu}{\nu}}\right)\left(r_t^2 + \frac{1}{\mu}\|\mathbf{x}_t - \mathbf{x}^*\|^2\right),$$

iterating for t, $t-1$, $\ldots 0$ we complete the proof. ■

8. Completing the Proofs of Main Results

In this section, we complete the proof of the main results, including Theorem 10 (solving general linear systems), Theorem 11 (solving positive definite linear systems) and Corollary 5 (solving linear systems with ℓ large singular values). We also discuss how the claims stated in Section 1 follow from these results.

Proof of Theorem 10 Recall that in Theorem 10 we consider an $m \times n$ matrix $\mathbf{A}$ with rank n, and a vector $\mathbf{b}$ such that the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is consistent. We then preprocess both the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ with a randomized Hadamard transform $\mathbf{Q}$ so that $\mathbf{A}$ is replaced by $\mathbf{Q}\mathbf{A}$ and $\mathbf{b}$ is replaced by $\mathbf{Q}\mathbf{b}$, which does not change the solution to the linear system. Then, Algorithm 1 is ran with $\mathbf{B} = \mathbf{I}$, and using the inexact PCG solve for step 6 as described in Lemma 24.

From Lemma 24, conditioned on an event $\mathcal{E}_t$ that holds with probability $1-\delta_{\mathrm{PCG}}$, in time $O(nk\log(k/\mu\delta_{\mathrm{PCG}}) + k^{\omega})$ PCG solves the t-th projection step to within sufficient accuracy so that we can apply our convergence analysis of inexact sketch-and-project from Lemma 23. Using Lemma 23, we can recover convergence in terms of $\|\mathbf{x}_t - \mathbf{x}^*\|^2$ error by converting from the error $\Delta_t = \|\mathbf{v}_t - \mathbf{x}^*\|^2_{\mathbb{E}[\mathbf{P}]^\dagger} + \frac{1}{\mu}\|\mathbf{x}_t - \mathbf{x}^*\|^2$, where recall that we define the projection matrix $\mathbf{P} = (\mathbf{S}\mathbf{A})^\dagger\mathbf{S}\mathbf{A}$ and denote its smallest eigenvalue as $\mu = \lambda_{\min}(\mathbb{E}[\mathbf{P}]) = \|\mathbb{E}[\mathbf{P}]^\dagger\|^{-1}$. Then, with the choice of α, β, γ as in Lemma 23,

$$\begin{aligned}\mathbb{E}[\Delta_{t+1}] &= \Pr(\mathcal{E}_t)\mathbb{E}[\Delta_{t+1} \mid \mathcal{E}_t] + \Pr(\neg\mathcal{E}_t)\mathbb{E}[\Delta_{t+1} \mid \neg\mathcal{E}_t] \\ &\le \Big(1 - \frac{1}{2}\sqrt{\frac{\mu}{\nu}}\Big)\mathbb{E}[\Delta_t] + \delta_{\mathrm{PCG}} \cdot \mathbb{E}[\Delta_{t+1} \mid \neg\mathcal{E}_t],\end{aligned}$$

where recall that $\nu = \lambda_{\max}(\mathbb{E}[(\bar{\mathbf{P}}^{-1/2}\mathbf{P}\bar{\mathbf{P}}^{-1/2})^2])$.

To handle the $\neg\mathcal{E}_t$ case, we use the second estimate of Lemma 24 which holds unconditionally and states that $\|\hat{\mathbf{w}}_t - \mathbf{w}_t\| \le 4\|\mathbf{w}_t\|$ where $\hat{\mathbf{w}}_t$ is the result of the steps 6 and 7 of Algorithm 1, performed inexactly using PCG. Then,

$$\|\hat{\mathbf{w}}_t\| \le \|\mathbf{w}_t\| + \|\hat{\mathbf{w}}_t - \mathbf{w}_t\| \le 5\|\mathbf{w}_t\| \le 5\|\mathbf{y}_t - \mathbf{x}^*\|,$$

where the last step follows since $\|\mathbf{w}_t\| = \|\mathbf{P}(\mathbf{y}_t - \mathbf{x}^*)\| \le \|\mathbf{y}_t - \mathbf{x}^*\|$. Using this, direct calculation shows that:

$$\begin{aligned}\Delta_{t+1} &\le \frac{1}{\mu}\|\mathbf{v}_{t+1} - \mathbf{x}^*\|^2 + \frac{1}{\mu}\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 \\ &\le \frac{1}{\mu}\Big(\|\mathbf{v}_t - \mathbf{x}^*\| + \|\mathbf{y}_t - \mathbf{x}^*\| + \gamma\|\hat{\mathbf{w}}_t\|\Big)^2 + \frac{1}{\mu}\Big(\|\mathbf{y}_t - \mathbf{x}^*\| + \|\hat{\mathbf{w}}_t\|\Big)^2 \\ &\le \frac{1}{\mu}\Big(\|\mathbf{v}_t - \mathbf{x}^*\| + (1+5\gamma)\|\mathbf{y}_t - \mathbf{x}^*\|\Big)^2 + \frac{6^2}{\mu}\|\mathbf{y}_t - \mathbf{x}^*\|^2 \\ &\le \frac{2}{\mu}\|\mathbf{v}_t - \mathbf{x}^*\|^2 + \frac{2(1+5\gamma)^2 + 6^2}{\mu}\|\mathbf{y}_t - \mathbf{x}^*\|^2.\end{aligned}$$

Now, using that $\|\mathbf{y}_t - \mathbf{x}^*\| \le \alpha\|\mathbf{v}_t - \mathbf{x}^*\| + \|\mathbf{x}_t - \mathbf{x}^*\|$, $\alpha\gamma \le 1$, and $\gamma^2 \le 1/\mu$, we obtain

$$\Delta_{t+1} = O\Big(\frac{1}{\mu}\|\mathbf{v}_t - \mathbf{x}^*\|^2 + \frac{1}{\mu^2}\|\mathbf{x}_t - \mathbf{x}^*\|^2\Big) \le \frac{C}{\mu}\Delta_t$$

for some constant $C > 0$. Thus, setting $\delta_{\mathrm{PCG}} = \frac{\mu^{3/2}}{4C\nu^{1/2}}$ and adjusting the runtime accordingly, we obtain $\mathbb{E}[\Delta_{t+1}] \leq (1 - \frac{1}{4}\sqrt{\mu/\nu})\mathbb{E}[\Delta_t]$. It follows that:

$$\begin{aligned}\mathbb{E}\big[\|\mathbf{x}_t - \mathbf{x}^*\|^2\big] \leq \mu\mathbb{E}\big[\Delta_t\big] &\leq \left(1 - \frac{1}{4}\sqrt{\frac{\mu}{\nu}}\right)^t \mu\Delta_0 \\ &= \left(1 - \frac{1}{4}\sqrt{\frac{\mu}{\nu}}\right)^t \big(\mu\|\mathbf{x}_0 - \mathbf{x}^*\|^2_{\mathbb{E}[\mathbf{P}]^\dagger} + \|\mathbf{x}_0 - \mathbf{x}^*\|^2\big) \\ &\leq 2\left(1 - \frac{1}{4}\sqrt{\frac{\mu}{\nu}}\right)^t \|\mathbf{x}_0 - \mathbf{x}^*\|^2,\end{aligned}$$

Now, we can use our first-order and second-order projection analysis to bound μ and ν respectively, obtaining:

$$\begin{aligned}&\text{(Corollary 13)} \qquad \mu \geq \frac{c_1\ell}{n\bar{\kappa}_\ell^2}, \\ &\text{(Corollary 21)} \qquad \nu \leq \frac{c_2 n}{\ell}\bar{\kappa}_{\ell:2k}^2,\end{aligned}$$

where recall that ℓ is a value that satisfies $\ell \geq \frac{ck}{\log k}$. Putting these together with the above convergence guarantee we obtain:

$$\mathbb{E}\big[\|\mathbf{x}_t - \mathbf{x}^*\|^2\big] \leq 2\left(1 - \frac{\ell\sqrt{c_1/c_2}}{4n\bar{\kappa}_\ell\bar{\kappa}_{\ell:2k}}\right)^t \|\mathbf{x}_0 - \mathbf{x}^*\|^2.$$

The overall time complexity of the algorithm includes the $O(mn\log m)$ cost of performing the randomized Hadamard transform plus the per-iteration cost that is dominated by $O(nks) = O(nk\log^4(n/\delta))$ for computing the sketch $\mathbf{SA}$ using a $k \times m$ LESS-uniform matrix $\mathbf{S}$ with s non-zeros per row, plus the $O(nk\log(k/\mu\delta_{\mathrm{PCG}}) + k^\omega) = O(nk\log(n\bar{\kappa}_\ell) + k^\omega)$ cost of preconditioning and solving the linear system in step 6. Thus, the overall runtime is:

$$O(mn\log m + t(nk(\log^4(n/\delta) + \log(n\bar{\kappa}_\ell)) + k^\omega)),$$

concluding the proof. ■

Next, we discuss how the above proof needs to be adapted for solving positive definite linear systems, in order to obtain the improved dependence on the condition numbers, thus proving Theorem 11.

Proof of Theorem 11 Recall that there are several key differences in how we implement Algorithm 1 when $\mathbf{A}$ is positive definite. First, the randomized Hadamard transform preprocessing step has to be done in a way that preserves the positive definite structure of the problem. To that end, we must apply the matrix $\mathbf{Q}$ symmetrically on both sides of $\mathbf{A}$, thus replacing the original system $\mathbf{Ax} = \mathbf{b}$ with:

$$\mathbf{QAQ}^\top\mathbf{x} = \mathbf{Qb}.$$

We note that, while the resulting linear system is not strictly equivalent to the original one, we can easily recover the original solution at the end by returning $\mathbf{Q}^\top\mathbf{x}$ instead of $\mathbf{x}$, which can be computed in additional $O(n \log n)$ time. For clarity of exposition, let us now denote the preprocessed matrix $\mathbf{QAQ}^\top$ as $\bar{\mathbf{A}}$, and the preprocessed vector $\mathbf{Qb}$ as $\bar{\mathbf{b}}$.

The next difference in the positive definite case is in the choice of matrix $\mathbf{B}$. While in Theorem 10 we effectively dropped the matrix $\mathbf{B}$ by using $\mathbf{B} = \mathbf{I}$, here we set $\mathbf{B} = \bar{\mathbf{A}}$. The main difference in the implementation of the algorithm resulting from this change is, as we discussed in Section 7, the fact that the linear system in step 6 is now $(\mathbf{S}\bar{\mathbf{A}}\mathbf{S}^\top)\mathbf{u}_t = \mathbf{S}\bar{\mathbf{A}}\mathbf{y}_t - \mathbf{S}\mathbf{b}$, which can be solved more easily than in the general case. Specifically, we first precompute $\mathbf{S}\bar{\mathbf{A}}$ in time $O(nks) = O(nk\log^4(n/\delta))$, then we compute $\mathbf{S}\bar{\mathbf{A}}\mathbf{S}^\top$ in time $O(k^2 s)$ and $\mathbf{S}\bar{\mathbf{A}}\mathbf{y}_t - \mathbf{S}\mathbf{b}$ in time $O(nk)$. Finally, to solve the system, we just need to invert the $k \times k$ matrix $\mathbf{S}\bar{\mathbf{A}}\mathbf{S}^\top$ in time $O(k^\omega)$.

Finally, we also need to adapt our convergence analysis to the $\mathbf{B} = \bar{\mathbf{A}}$ case. Here, we rely on the original analysis of exact sketch-and-project from Gower et al. (2018) (see Remark 6). Using the notation from the remark, similarly as in the proof of Theorem 10 we can show:

$$\mathbb{E}\big[\|\mathbf{x}_t - \mathbf{x}^*\|_{\mathbf{B}}^2\big] \leq \bigg(1 - \sqrt{\frac{\mu}{\nu}}\bigg)^t \mu\Delta_0 \leq 2\bigg(1 - \sqrt{\frac{\mu}{\nu}}\bigg)^t \|\mathbf{x}_0 - \mathbf{x}^*\|_{\mathbf{B}}^2,$$

where μ and ν are computed with respect to the sketched projection matrix defined for the matrix $\bar{\mathbf{A}}\mathbf{B}^{-1/2}$. Since $\mathbf{B} = \bar{\mathbf{A}} = \mathbf{QAQ}^\top$, then $\bar{\mathbf{A}}\mathbf{B}^{-1/2} = \bar{\mathbf{A}}^{1/2} = \mathbf{Q}\mathbf{A}^{1/2}\mathbf{Q}^\top$, so we can write the projection matrix as:

$$\begin{aligned}\mathbf{P} &= \bar{\mathbf{A}}^{1/2}\mathbf{S}^\top(\mathbf{S}\bar{\mathbf{A}}\mathbf{S}^\top)^{-1}\mathbf{S}\bar{\mathbf{A}}^{1/2} \\ &= \mathbf{Q}\mathbf{A}^{1/2}\mathbf{Q}\mathbf{S}^\top(\mathbf{SQAQS}^\top)^{-1}\mathbf{SQ}\mathbf{A}^{1/2}\mathbf{Q}^\top = \mathbf{Q}\mathbf{P}'\mathbf{Q}^\top,\end{aligned}$$

where $\mathbf{P}' := \mathbf{A}^{1/2}\mathbf{Q}\mathbf{S}^\top(\mathbf{SQAQS}^\top)^{-1}\mathbf{SQ}\mathbf{A}^{1/2}$ is also a projection matrix. Note that applying the orthogonal matrix $\mathbf{Q}$ on both sides of $\mathbf{P}'$ does not affect the calculation of μ and ν, so we can equivalently compute them from the matrix $\mathbf{P}'$. Fortunately, this matrix is precisely the sketched projection we analyze in Corollaries 13 and 21 if we replace matrix $\mathbf{A}$ with matrix $\mathbf{A}^{1/2}$. Thus, in our calculations, the squared singular values become the (non-squared) eigenvalues, and we obtain the corresponding bounds:

$$\begin{aligned}&\text{(adapted Corollary 13)} \qquad \mu \geq \frac{c_1\ell}{n\bar{\kappa}_\ell^2(\mathbf{A}^{1/2})} = \frac{c_1\ell}{n\tilde{\kappa}_\ell}, \\ &\text{(adapted Corollary 21)} \qquad \nu \leq \frac{c_2 n}{\ell}\bar{\kappa}_{\ell:2k}^2(\mathbf{A}^{1/2}) = \frac{c_2 n}{\ell}\tilde{\kappa}_{\ell:2k}.\end{aligned}$$

The rest of the analysis proceeds same as in the general case, except with these better condition numbers. ■

We next show how to obtain Theorem 1 from Theorem 10. The main issue here is to translate the somewhat complex condition numbers appearing in our convergence rates into the simple condition number $\kappa_\ell = \sigma_\ell/\sigma_n$, and then perform the time complexity analysis of the resulting algorithm. Almost identical arguments (omitted here) can be applied to recover Theorem 2 from Theorem 11.

Proof of Theorem 1 Suppose that our sketch size satisfies $k = \tilde{O}(n^{\frac{1}{\omega-1}})$ and $\ell = \Omega(k/\log k)$. Then, recall that Theorem 10 implies:

$$\tilde{O}\Big(n^2 + \bar{\kappa}_\ell\bar{\kappa}_{\ell:2k}\frac{n}{\ell}(nk + k^\omega)\Big) = \tilde{O}\Big(\bar{\kappa}_\ell\bar{\kappa}_{\ell:2k}(n^2 + nk^{\omega-1})\Big) = \tilde{O}\big(\bar{\kappa}_\ell\bar{\kappa}_{\ell:2k}n^2\big).$$

We will now show that, with the right choice of k, we can bound the condition numbers $\bar{\kappa}_\ell\bar{\kappa}_{\ell:2k}$ by $\tilde{O}(\kappa_\ell)$, where $\kappa_\ell = \frac{\sigma_\ell}{\sigma_n}$. We will also use the shorthand $\kappa_{\ell:q} = \frac{\sigma_\ell}{\sigma_q}$. We use the following bound:

$$\begin{aligned}\bar{\kappa}_\ell\bar{\kappa}_{\ell:2k} &= \sqrt{\frac{\sum_{i>\ell}\sigma_i^2}{(n-\ell)\sigma_n^2}\bar{\kappa}_{\ell,2k}^2} \leq \sqrt{\frac{(2k-\ell)\sigma_\ell^2+(n-2k)\sigma_{2k}^2}{(n-\ell)\sigma_n^2}\,\frac{\sigma_\ell^2}{\sigma_{2k}^2}} \\ &= \frac{\sigma_\ell}{\sigma_n}\sqrt{\frac{2k-\ell}{n-\ell}\,\frac{\sigma_\ell^2}{\sigma_{2k}^2}+\frac{n-2k}{n-\ell}} \leq \kappa_\ell\sqrt{\frac{2k}{n}\kappa_{\ell:2k}^2+1}, \end{aligned} \tag{17}$$

where we used the fact that $\bar{\kappa}_{\ell,2k} \leq \kappa_{\ell,2k}$. Thus, as long as the term under the square root is $\tilde{O}(1)$, then we have obtained the desired time complexity $\tilde{O}(\kappa_\ell n^2)$ with $\ell = \Omega(n^{\frac{1}{\omega-1}})$. Naturally, this may not be true for some choices of k, namely if there is a sharp drop in the singular values in the range from σ_ℓ to σ_{2k}. To address this, we devise an iterative procedure that always finds a value of k that avoids this phenomenon.

Recall that Theorem 10 uses $\ell = \lceil\frac{c_2 k}{\log k}\rceil$, which means that the ratio of $\frac{2k}{\ell}$ can be bounded by $b := C\log k$. Let us start with an initial choice of k being some $k_0 = \lceil n^{\frac{1}{\omega-1}}\log n\rceil$ and the corresponding initial value of ℓ equal to $\ell_0 = \lceil 2k_0/b\rceil = \Omega(n^{\frac{1}{\omega-1}})$. First, note that we can assume without loss of generality that $\kappa_{\ell_0} \leq n^{\omega-2}$, because otherwise the desired complexity $\tilde{O}(\kappa_\ell n^2)$ can be obtained by using a direct $O(n^\omega)$ time solver. Now, consider the following procedure, that iterates over $q = 0, 1, ...$:

1. If $\kappa_{\ell_q:2k_q} \leq \sqrt{n/k_0}$ or $\kappa_{\ell_q}^2 \leq \kappa_{\ell_0}$, then stop and choose $k = k_q$.
2. Otherwise, set $k_{q+1} = \lceil bk_q\rceil$, with $\ell_{q+1} = \lceil 2k_{q+1}/b\rceil$, and repeat the procedure with $q = q+1$.

Now, observe that after each iteration in which we do not stop, we have $\kappa_{\ell_q:2k_q} > \sqrt{n/k_0}$, which means that:

$$\kappa_{\ell_{q+1}} = \frac{\sigma_{\ell_{q+1}}}{\sigma_n} = \frac{\sigma_{\ell_q}}{\sigma_n}\,\frac{\sigma_{\ell_{q+1}}}{\sigma_{\ell_q}} \leq \kappa_{\ell_q}\frac{\sigma_{2k_q}}{\sigma_{\ell_q}} = \frac{\kappa_{\ell_q}}{\kappa_{\ell_q:2k_q}} \leq \sqrt{k_0/n}\cdot\kappa_{\ell_q},$$

where we used that fact that $\ell_{q+1} \geq 2k_q$, and therefore $\sigma_{\ell_{q+1}} \leq \sigma_{2k_q}$. So, going from q to $q+1$ we decrease the condition number κ_{ℓ_q} by at least a factor of:

$$\sqrt{k_0/n} = \tilde{O}(n^{\frac{1}{2}(\frac{1}{\omega-1}-1)}\log^{1/2} n) = \tilde{O}(n^{-\frac{\omega-2}{2(\omega-1)}}\log^{1/2} n).$$

This means that after q iterations of this procedure, we get:

$$\kappa_{\ell_q}^2 \leq \kappa_{\ell_0}^2 n^{-q\frac{\omega-2}{\omega-1}}\log^q n \leq \kappa_{\ell_0} n^{(\omega-2)(1-\frac{q}{\omega-1})}\log^q n,$$

where we used the assumption that $\kappa_{\ell_0} = O(n^{\omega-2})$. Now, note that since $\omega - 1 < 2$, we have $n^{(\omega-2)(1-\frac{q}{\omega-1})} < n^{-\theta}$ for some $\theta > 0$ for any $q \geq 2$. So already after $q \leq 2$ iterations, the stopping condition $\kappa_{\ell_q}^2 \leq \kappa_{\ell_0}$ must occur. If it does, then we get $\bar{\kappa}_{\ell_q}\bar{\kappa}_{\ell_q:2k_q} \leq \kappa_{\ell_q}^2 \leq \kappa_{\ell_0}$, which results in the desired time complexity $\tilde{O}(\kappa_{\ell_0} n^2)$. On the other hand, if the other stopping condition occurs, i.e., $\kappa_{\ell_q:2k_q} \le \sqrt{n/k_0}$, then we can use the bound in (17) to get

$$\bar{\kappa}_{\ell_q}\bar{\kappa}_{\ell_q:2k_q} \le \kappa_{\ell_q}\sqrt{\frac{2k_q}{n}\kappa^2_{\ell_q:2k_q}+1} \le \kappa_{\ell_q}\sqrt{\frac{2k_q}{n}\frac{n}{k_0}+1} = O\big(\kappa_{\ell_0}\cdot\log^{q/2} k\big).$$

Since $q \le 2$, we again recover the time complexity bound $\tilde{O}(\kappa_{\ell_0} n^2)$. In other words, this procedure shows that in the worst case we only need to increase the sketch size k by a factor of $O(\log^2 n)$ to find the value that will lead to the desired time complexity. ■

Next, we describe how our improved DPP coupling argument used in Lemma 12 allows us to give a direct improvement in the time complexity of solving linear systems with ℓ large singular values, compared to the previous result from Dereziński and Yang (2024).

Proof of Corollary 5 Consider an $n \times n$ linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ such that all of the singular values of matrix $\mathbf{A}$ except for the top ℓ are within a constant factor of each other, i.e., $\sigma_{\ell+1}(\mathbf{A}) = O(\sigma_n(\mathbf{A}))$, which is the setting from Dereziński and Yang (2024). In particular, this implies that for this matrix $\bar{\kappa}_\ell \le C = O(1)$. Let us consider a slightly different implementation of Algorithm 1, where instead of a LESS-uniform sketching matrices $\mathbf{S}$ with $s \ge C\log^4(n/\delta)$ non-zeros per row, we are going to use a block sub-sampling matrix (equivalent to taking LESS-uniform with $s = 1$). Crucially, our first-order projection analysis (Lemma 12) still applies in this case, showing that:

$$\text{(Corollary 13)} \qquad \mu \ge \frac{c_1\ell}{n\bar{\kappa}_\ell^2} \ge \frac{c_1\ell}{C^2 n} \quad \text{for} \quad k = O(\ell\log\ell).$$

While our second-order projection analysis does not apply for a block sub-sampling matrix, here we can use a simpler upper bound on ν, which was given in Lemma 2 of Gower et al. (2018), stating that $\nu \le 1/\mu$. This gives a convergence guarantee of the form $\mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^*\|^2] \le 2(1-\mu)^t\|\mathbf{x}_0 - \mathbf{x}^*\|^2$, and the resulting iteration complexity is $t = O(\mu^{-1}\log(1/\epsilon)) = O(\frac{n}{\ell}\log(1/\epsilon))$. Note that, for this variant of the algorithm, the per iteration time complexity is $O(nk)$ to construct the sketch $\mathbf{SA}$, and $O(nk\log(n) + k^\omega)$ to solve the projection step 6. Thus, we get the following overall time complexity:

$$O\Big(n^2\log n + \frac{n}{\ell}(nk\log(n) + k^\omega)\log 1/\epsilon\Big) = O\Big((n^2\log^2 n + n\ell^{\omega-1}\log^\omega \ell\Big)\log 1/\epsilon\Big).$$

We note that the result previously obtained for this problem by Dereziński and Yang (2024) relied on a different DPP coupling argument, which requires $k = O(\ell\log^3 n)$. We improve this to $k = O(\ell\log\ell)$, leading to better logarithmic factors in the time complexity. ■

Extension to sparse least squares. Finally, we discuss how our algorithms can be adapted and applied in the tall and sparse least squares setting. Recall that in the least squares task, we are given a tall $n \times d$ matrix $\mathbf{A}$ with $\mathrm{nnz}(\mathbf{A})$ non-zero entries, and an n-dimensional vector $\mathbf{b}$, and our goal is to minimize $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$. This can be achieved by running preconditioned gradient descent on f, which takes the form of $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\hat{\mathbf{H}}^{-1}\mathbf{g}_t$, where $\mathbf{g}_t = \nabla f(\mathbf{x}_t) = \mathbf{A}^\top(\mathbf{A}\mathbf{x}_t - \mathbf{b})$ is the gradient, which can be computed in $O(\mathrm{nnz}(\mathbf{A}))$ time, whereas $\hat{\mathbf{H}}$ is a preconditioner which approximates the Hessian matrix $\nabla^2 f(\mathbf{x}_t) = \mathbf{A}^\top\mathbf{A}$. This preconditioner can be constructed by computing an $\tilde{O}(d) \times d$ sketch $\tilde{\mathbf{A}} = \mathbf{\Phi}\mathbf{A}$ where $\mathbf{\Phi}$ is a sparse oblivious subspace embedding (similarly as in Lemma 24; e.g., see Chenakkod et al., 2024), and defining $\hat{\mathbf{H}} = \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}}$. Now, instead of computing $\hat{\mathbf{H}}^{-1}$ outright at the cost of $\tilde{O}(d^\omega)$, we use Algorithm 1 to solve the linear system $\hat{\mathbf{H}}\mathbf{x} = \mathbf{g}_t$ at every iteration. Since $\hat{\mathbf{H}}$ is positive definite, we can use the version considered in Theorem 11, however this has to be further modified to account for the fact that we can never actually form this matrix, but rather must operate on its decomposition $\hat{\mathbf{H}} = \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}}$. Fortunately, this *implicit* positive definite system solver is a simple adaptation of Sketch-and-Project (see Section 9 of Dereziński and Yang, 2024), which is not affected by Nesterov's acceleration. In the end, the overall time complexity of this procedure includes $\tilde{O}(\mathrm{nnz}(\mathbf{A}))$ for computing $\tilde{\mathbf{A}}$, and $\tilde{O}(\mathrm{nnz}(\mathbf{A}) + d^2\kappa_\ell)$ in each iteration of the preconditioned gradient descent, for computing the gradient and then running the solver, respectively, where $\kappa_\ell = \sigma_\ell(\mathbf{A})/\sigma_d(\mathbf{A})$. Since the resulting algorithm only requires $O(\log 1/\epsilon)$ steps to converge, the final time complexity is $\tilde{O}((\mathrm{nnz}(\mathbf{A}) + d^2\kappa_\ell)\log 1/\epsilon)$.

9. Lower bound for Matrix-vector Query Algorithms

In this section, we give lower bounds for a class of linear system solvers, which include conjugate gradient as well as certain randomized preconditioned solvers, in our fine-grained setting where the linear system has a bounded spectral tail condition number κ_ℓ. We build on existing lower bounds for solving positive definite systems, given by Braverman et al. (2020), who obtained them with respect to the classical condition number κ. The model of computation we consider includes all algorithms which access the matrix $\mathbf{A}$ through matrix-vector queries of the form $\mathbf{A}\mathbf{v}$, where the vector $\mathbf{v}$ can be randomized and chosen adaptively.

Definition 25 (Matrix-vector query model, Braverman et al. (2020)) *We say that* Alg *is a* MatVec *algorithm for solving positive definite linear systems if, given initial point* $\mathbf{x}_0 \in \mathbb{R}^n$ *and* $\mathbf{b} \in \mathbb{R}^n$*, it interacts with a positive definite matrix* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *via* T *adaptive randomized queries,* $\mathbf{w}_t = \mathbf{A}\mathbf{v}_t$*, and returns an estimate* $\tilde{\mathbf{x}} \in \mathbb{R}^n$ *of* $\mathbf{A}^{-1}\mathbf{b}$*. We call* T *the query complexity of* Alg*.*

This computation model includes many standard deterministic iterative algorithms, most notably conjugate gradient (CG), and it also allows for certain preconditioning techniques, for example using Randomized SVD to build a preconditioner that approximates the top-ℓ part of the spectrum of $\mathbf{A}$. The central question of this section is:

> *Given* n*,* $\ell < n$ *and* $\kappa_\ell \geq 1$*, what is the* MatVec *query complexity of solving an* $n \times n$ *positive definite linear system* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *with* $\frac{\sigma_\ell(\mathbf{A})}{\sigma_n(\mathbf{A})} \leq \kappa_\ell$*?*

As discussed in Section 1, CG can solve this problem using $O(\ell + \sqrt{\kappa_\ell}\log 1/\epsilon)$ MatVec queries. An alternative strategy is to use a preconditioned solver. To do this, we can probe the matrix $\mathbf{A}$ using $\tilde{O}(\ell)$ Gaussian random vector queries to construct a rank ℓ approximation via the Randomized SVD algorithm with power iteration (Halko et al., 2011). Augmented with a preconditioner based on such an approximation, CG can solve the system using only $O(\sqrt{\kappa_\ell}\log 1/\epsilon)$ queries. Even though the latter strategy uses randomization, while the former is fully deterministic, the overall query complexity is still no better than CG (although preconditioning is often preferred in practice due to its improved stability properties). Can we achieve a better query complexity in the MatVec model for our problem?

Next, we show that the guarantee attained by CG is in fact essentially optimal among all MatVec algorithms (even allowing randomization), up to logarithmic factors. In particular, this means that for any $\ell = \Omega(n^\theta)$, where $\theta > 0$, no MatVec algorithm has time complexity $\tilde{O}(\sqrt{\kappa_\ell} \cdot n^2 \log 1/\epsilon)$ for dense positive definite linear systems, yet we show this for Sketch-and-Project with Nesterov's acceleration (Theorem 2) given any $\ell = O(n^{0.729})$.

Theorem 26 *Any* MatVec *algorithm that, given* $\mathbf{x}_0$ *and an* $n \times n$ *positive definite linear system* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *with* $\frac{\sigma_\ell(\mathbf{A})}{\sigma_n(\mathbf{A})} \leq \kappa_\ell$ *returns* $\tilde{\mathbf{x}}$ *such that:*

$$\Pr\Big(\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_{\mathbf{A}}^2 \leq \epsilon \|\mathbf{A}\| \|\mathbf{x}_0 - \mathbf{x}^*\|^2\Big) \geq 1 - \frac{1}{e}, \quad \textit{where } \mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b},$$

for $\epsilon \leq \frac{1}{e}\min\{\ell^{-2}, \kappa_\ell^{-1}\}$*, must have query complexity at least* $\tilde{\Omega}(\ell + \sqrt{\kappa_\ell})$*.*

Remark 7 *If we additionally assume that the* MatVec *algorithm is deterministic, then we can obtain a stronger query complexity lower bound of* $\Omega(\ell + \sqrt{\kappa_\ell}\log 1/\epsilon)$*, which matches the CG upper bound down to constant factors.*

To establish the above result, we actually provide a more general reduction which shows how to obtain a lower bound for our fine-grained linear system task by combining two types of existing lower bounds: one expressed in terms of the dimension of the problem (without restricting the spectrum), and one expressed in terms of the overall condition number of the input matrix.

Lemma 27 *Let* $\mathcal{A}$ *denote some family of* MatVec *algorithms, and let* $\mathcal{L}$ *denote the family of square positive definite linear system tasks* $(\mathbf{A}, \mathbf{b}, \mathbf{x}_0)$*, where* $\mathbf{x}_0$ *is the starting point. Define* $T_{\mathcal{A},\mathcal{L}}(n, \kappa, \epsilon, \delta)$ *as the minimum query complexity among algorithms in* $\mathcal{A}$ *that solve all* $n \times n$ *linear systems in* $\mathcal{L}$ *such that* $\kappa(\mathbf{A}) \leq \kappa$*, so that*

$$\Pr\Big(\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_{\mathbf{A}}^2 \leq \epsilon \|\mathbf{A}\| \|\mathbf{x}_0 - \mathbf{x}^*\|^2\Big) \geq 1 - \delta, \quad \textit{where } \mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}. \tag{18}$$

Now, let T *denote the best query complexity among algorithms in* $\mathcal{A}$ *that solve all* $n \times n$ *linear systems in* $\mathcal{L}$ *with spectral tail condition number* $\frac{\sigma_\ell(\mathbf{A})}{\sigma_n(\mathbf{A})} \leq \kappa_\ell$*, in the sense of* (18)*. Then,*

$$T \geq \max\Big\{T_{\mathcal{A},\mathcal{L}}(\ell, \infty, \epsilon, \delta), T_{\mathcal{A},\mathcal{L}}(n - \ell, \kappa_\ell, \epsilon, \delta)\Big\}.$$

Remark 8 *An analogous reduction can be obtained along the same lines if we replace* $\mathcal{L}$ *with the family of all square linear systems (dropping the positive definiteness).*

Proof First, let us use $\mathcal{L}_\ell(n, \kappa_\ell)$ to denote the above defined family of $n \times n$ linear systems from $\mathcal{L}$ with bounded spectral tail condition number. We break the proof down into two cases, depending on which of the two terms dominates the value of the max in the lower bound.

Case 1: $T_{\mathcal{A},\mathcal{L}}(\ell, \infty, \epsilon, \delta) \geq T_{\mathcal{A},\mathcal{L}}(n - \ell, \kappa_\ell, \epsilon, \delta)$. We proceed via a proof by contradiction. Suppose that some $\mathsf{Alg} \in \mathcal{A}$ has query complexity less than $T_{\mathcal{A},\mathcal{L}}(\ell, \infty, \epsilon, \delta)$ for solving $\mathcal{L}_\ell(n, \kappa_\ell)$. We are going to show how to use Alg to solve *all* $\ell \times \ell$ linear systems in $\mathcal{L}$, thereby obtaining a contradiction. Suppose that $(\mathbf{A}, \mathbf{b}, \mathbf{x}_0)$ is an $\ell \times \ell$ linear system in $\mathcal{L}$, and define:

$$(\bar{\mathbf{A}}, \bar{\mathbf{b}}, \bar{\mathbf{x}}_0) = \left(\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \sigma_\ell(\mathbf{A})\mathbf{I}_{n-\ell} \end{bmatrix}, \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{0} \end{bmatrix} \right).$$

Note that $(\bar{\mathbf{A}}, \bar{\mathbf{b}}, \bar{\mathbf{x}}_0) \in \mathcal{L}_\ell(n, \kappa_\ell)$, since $\bar{\mathbf{A}}$ is positive definite and $\frac{\sigma_\ell(\bar{\mathbf{A}})}{\sigma_n(\bar{\mathbf{A}})} = 1 \le \kappa_\ell$. Moreover, if Alg runs on this system and returns $\bar{\mathbf{x}}$ such that $\|\bar{\mathbf{x}} - \bar{\mathbf{x}}^*\|_{\bar{\mathbf{A}}}^2 \le \epsilon \|\bar{\mathbf{A}}\| \|\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}^*\|^2$, where $\bar{\mathbf{x}}^* = \bar{\mathbf{A}}^{-1}\bar{\mathbf{b}}$, then the vector $\tilde{\mathbf{x}}$ consisting of the first ℓ coordinates of $\bar{\mathbf{x}}$ satisfies $\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_{\mathbf{A}}^2 \le \epsilon \|\mathbf{A}\| \|\mathbf{x}_0 - \mathbf{x}^*\|^2$, where $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$. This gives us the contradiction.

Case 2: $T_{\mathcal{A},\mathcal{L}}(\ell, \infty, \epsilon, \delta) > T_{\mathcal{A},\mathcal{L}}(n-\ell, \kappa_\ell, \epsilon, \delta)$. This case follows similarly. Suppose that some $\mathsf{Alg} \in \mathcal{A}$ has query complexity less than $T_{\mathcal{A},\mathcal{L}}(n-\ell, \kappa_\ell, \epsilon, \delta)$ for solving $\mathcal{L}_\ell(n, \kappa_\ell)$. Now, consider an $(n-\ell) \times (n-\ell)$ linear system task $(\mathbf{A}, \mathbf{b}, \mathbf{x}_0) \in \mathcal{L}$ with condition number $\kappa(\mathbf{A}) \le \kappa_\ell$, and define:

$$(\bar{\mathbf{A}}, \bar{\mathbf{b}}, \bar{\mathbf{x}}_0) = \left(\begin{bmatrix} \sigma_1(\mathbf{A})\mathbf{I}_\ell & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{x}_0 \end{bmatrix} \right).$$

Note that, as in Case 1, we have $(\bar{\mathbf{A}}, \bar{\mathbf{b}}, \bar{\mathbf{x}}_0) \in \mathcal{L}_\ell(n, \kappa_\ell)$, since $\bar{\mathbf{A}}$ is positive definite and $\frac{\sigma_\ell(\bar{\mathbf{A}})}{\sigma_n(\bar{\mathbf{A}})} = \frac{\sigma_1(\mathbf{A})}{\sigma_{n-\ell}(\mathbf{A})} = \kappa(\mathbf{A}) \le \kappa_\ell$. Now, solving this system to ϵ accuracy allows us to recover an ϵ approximate solution for $(\mathbf{A}, \mathbf{b}, \mathbf{x}_0)$, which is a contradiction with the definition of $T_{\mathcal{A},\mathcal{L}}(n-\ell, \kappa_\ell, \epsilon, \delta)$. This concludes the proof of the lemma. ∎

To complete the proof of Theorem 26, we will rely on the following lower bound for MatVec algorithms, given by Braverman et al. (2020), which is shown via a reduction from the task of finding the largest eigenvector of a matrix.

Lemma 28 (Theorem 2.3, Braverman et al. (2020)) *For all $d \ge c$ and $s \in [c, d]$, where c is a sufficiently large absolute constant, any MatVec algorithm which satisfies the guarantee:*

$$\Pr\left(\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_{\mathbf{A}}^2 \le \frac{1}{e} \frac{\|\mathbf{A}\| \|\mathbf{x}_0 - \mathbf{x}^*\|^2}{s^2} \right) \ge 1 - \frac{1}{e}, \quad \textit{where } \mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b},$$

for all $(d+s^2)$-sparse $d \times d$ positive definite matrices $\mathbf{A}$ with $\kappa(\mathbf{A}) \le s^2$, and all $\mathbf{x}_0, \mathbf{b} \in \mathbb{R}^d$, must have query complexity at least $\tilde{\Omega}(s)$.

We are now ready to complete the proof of Theorem 26.

Proof of Theorem 26 Thanks to Lemma 27, it suffices to lower bound $T_{\mathcal{A},\mathcal{L}}(\ell, \infty, \epsilon, \delta)$ and $T_{\mathcal{A},\mathcal{L}}(n-\ell, \kappa_\ell, \epsilon, \delta)$, with $\epsilon = \frac{1}{e}\min\{\ell^{-2}, \kappa_\ell^{-1}\}$ and $\delta = 1/e$. For the former, we use Lemma 28 with $d = \ell$ and $s = \ell - 1$, obtaining that $T_{\mathcal{A},\mathcal{L}}(\ell, \infty, \epsilon, 1/e) = \tilde{\Omega}(\ell)$. On the other hand, for the latter, we use Lemma 28 with $d = n - \ell$ and $s = \sqrt{\kappa_\ell}$ (without loss of generality, we can assume that $\sqrt{\kappa_\ell} \le n - \ell$), obtaining that $T_{\mathcal{A},\mathcal{L}}(n-\ell, \kappa_\ell, \epsilon, \delta) = \tilde{\Omega}(\sqrt{\kappa_\ell})$. Putting these two lower bounds together, we obtain a complexity lower bound of the form $\max\{\tilde{\Omega}(\ell), \tilde{\Omega}(\sqrt{\kappa_\ell})\} = \tilde{\Omega}(\ell + \sqrt{\kappa_\ell})$. ∎

Finally, to recover the claim in Remark 7, we observe that for deterministic MatVec algorithms one can rely on classical lower bounds developed by Nemirovsky and Yudin (1983) (e.g., see Theorem 7.2.6), which show that $T_{\mathcal{A},\mathcal{L}}(n, \kappa, \epsilon, \delta) = \Omega(\min\{n, \sqrt{\kappa}\log 1/\epsilon\})$ (here, since the algorithm is deterministic, δ can be any positive probability).

10. Conclusions

In this paper, we developed a nuanced framework for analyzing iterative linear system solvers that allows us to obtain sharper convergence guarantees than classical perspectives. By introducing a more flexible condition number κ_ℓ that depends on the tail of the spectrum of the matrix, we provided a finer-grained analysis than traditional approaches that rely on a single condition number. Our stochastic algorithm, based on the Sketch-and-Project paradigm, provides improved convergence guarantees for solving linear systems in many machine learning settings, particularly for large matrices where the top portion of the spectrum is ill-conditioned, while the tail is controlled. We highlighted the significance of these improvements for models such as spiked covariance and kernel ridge regression, where low-rank structure is likely. In addition, we demonstrated a clear separation between the performance of stochastic solvers and traditional matrix-vector product-based methods like the well-known preconditioned conjugate gradient.

Acknowledgments

The work was partially supported by NSF grant CCF-2338655 (MD), ARO grant 2003514594 (DL), NSF grant DMS-2011140 (DN), and NSF grant DMS-2309685 (ER).

Appendix A. Gaussian Universality of Sketched Isometric Embeddings

In this section, we give the proof of Lemma 9, which bounds the extreme singular values of the random matrix $\mathbf{SU}$, where $\mathbf{S}$ is a $k \times m$ LESS-uniform embedding (see Definition 6) and $\mathbf{U}$ is an $m \times d$ matrix such that $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$ (an isometric embedding) with row norms of $\mathbf{U}$ bounded by $C\sqrt{d/m}$ for some absolute constant C. Essentially, we show that as long as $\mathbf{S}$ has $O(\log^4(d/\delta))$ non-zeros per row, then the extreme singular values of $\mathbf{SU}$ behave just like the extreme singular values of the corresponding Gaussian matrix.

Our Gaussian universality analysis of the random matrix $\mathbf{SU}$ follows similarly to the analysis of Chenakkod et al. (2024), who showed that such matrices are subspace embeddings when $k \geq 2d$. Here, the key difference is that we consider the setting where $\mathbf{SU}$ is a wide matrix (because $k < d$), which means that it cannot be a subspace embedding. Yet the Gaussian universality analysis can still be applied to bound the extreme singular values.

This analysis is based on a universality result of Brailovskaya and van Handel (2024), which compares the spectrum of a sum of independent random matrices to that of a Gaussian random matrix with the same mean and covariance structure. In the following, we use $\mathrm{spec}(\mathbf{X})$ to denote the spectrum of matrix $\mathbf{X}$, and if $\mathbf{X}$ is a random $d \times d$ matrix, we let $\mathrm{Cov}(\mathbf{X})$ be the $d^2 \times d^2$ covariance matrix of the entries of $\mathbf{X}$. Finally, we define the Hausdorff distance between two subsets $\mathcal{A}, \mathcal{B} \subseteq \mathbb{R}$ as

$$d_H(\mathcal{A}, \mathcal{B}) := \inf\{\epsilon > 0 : \mathcal{A} \subseteq \mathcal{B} + [-\epsilon, \epsilon] \text{ and } \mathcal{B} \subseteq \mathcal{A} + [-\epsilon, \epsilon]\}.$$

Lemma 29 (Theorem 2.4, Brailovskaya and van Handel (2024)) *Given a random matrix model* $\mathbf{X} := \mathbf{Z}_0 + \sum_{i=1}^n \mathbf{Z}_i$*, where* $\mathbf{Z}_0$ *is a symmetric deterministic* $d \times d$ *matrix* *and* $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$ *are symmetric independent random matrices with* $\mathbb{E}[\mathbf{Z}_i] = 0$ *and* $\|\mathbf{Z}_i\| \le R$, *define the following:*

$$\sigma(\mathbf{X}) = \left\|\mathbb{E}[(\mathbf{X} - \mathbb{E}\mathbf{X})^2]\right\|^{\frac{1}{2}} \quad \text{and} \quad \sigma_*(\mathbf{X}) = \sup_{\|\mathbf{v}\|=\|\mathbf{w}\|=1} \mathbb{E}\Big[|\mathbf{v}^\top(\mathbf{X} - \mathbb{E}\mathbf{X})\mathbf{w}|^2\Big]^{\frac{1}{2}}.$$

Let $\mathbf{G}$ *be a* $d \times d$ *symmetric random matrix with jointly Gaussian entries, such that* $\mathbb{E}[\mathbf{G}] = \mathbb{E}[\mathbf{X}]$ *and* $\mathrm{Cov}(\mathbf{G}) = \mathrm{Cov}(\mathbf{X})$. *There is a universal constant* $C > 0$ *such that for any* $t \ge 0$,

$$\Pr\Big(d_H\big(\mathrm{spec}(\mathbf{X}), \mathrm{spec}(\mathbf{G})\big) > C\epsilon(t)\Big) \le de^{-t},$$
$$\text{where} \quad \epsilon(t) = \sigma_*(\mathbf{X})t^{\frac{1}{2}} + R^{\frac{1}{3}}\sigma(\mathbf{X})^{\frac{2}{3}}t^{\frac{2}{3}} + Rt. \tag{19}$$

A result from Chenakkod et al. (2024) allows us to bound the variance parameters σ_* and σ appearing in Lemma 29. We will apply Lemma 29 to a symmetrized version of the $\mathbf{SU}$ matrix. For any rectangular matrix $\mathbf{A}$, define

$$\mathrm{sym}(\mathbf{A}) := \begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^\top & 0 \end{bmatrix}.$$

Lemma 30 (Lemma 3.5, Chenakkod et al. (2024), Covariance Parameters) *Let* $\mathbf{S} = \{s_{ij}\}_{i\in[k],j\in[m]}$ *be a* $k \times m$ *random matrix such that* $\mathbb{E}(s_{ij}) = 0$ *and* $\mathrm{Var}(s_{ij}) = p$ *for all* $i \in [k], j \in [m]$, *and* $\mathrm{Cov}(s_{ij}, s_{k\ell}) = 0$ *for any* $\{i, t\} \subset [k], \{j, \ell\} \subset [m]$ *and* $(i, j) \ne (t, \ell)$. *Let* $\mathbf{U}$ *be an arbitrary deterministic matrix such that* $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$. *We then have*

$$\sigma_*(\mathrm{sym}(\mathbf{SU})) \le 2\sqrt{p} \quad \text{and} \quad \sigma(\mathrm{sym}(\mathbf{SU})) \le \sqrt{pk}.$$

We are now ready to give a proof of Lemma 9.

Proof of Lemma 9 We are going to apply the universality result to the following matrix, where we use $\mathbf{u}_i^\top$ to denote the ith row of $\mathbf{U}$, and $\mathbf{e}_i$ to denote the ith standard basis vector:

$$\mathbf{X} = \mathrm{sym}(\mathbf{SU}) = \sum_{i\in[k],j\in[s]} \underbrace{\mathrm{sym}\Big(\sqrt{\frac{m}{s}}r_{ij}\mathbf{e}_i\mathbf{u}_{I_{ij}}^\top\Big)}_{\mathbf{Z}_{ij}},$$

It is easy to verify that the sketching matrix $\mathbf{S} = \{s_{ij}\}_{i\in[k],j\in[m]}$ from Definition 6 satisfies $\mathbb{E}[s_{ij}] = 0$, $\mathrm{Var}[s_{ij}] = 1$ and $\mathrm{Cov}(s_{ij}, s_{t\ell}) = 0$ for any $\{i, t\} \subset [k], \{j, \ell\} \subset [m]$ and $(i, j) \ne (t, \ell)$. Applying Lemma 30 with $p = 1$, we have $\sigma_*(\mathrm{sym}(\mathbf{SU})) \le 2$ and $\sigma(\mathrm{sym}(\mathbf{SU})) \le \sqrt{k}$. Moreover, using that $\|\mathbf{u}_i\| \le C\sqrt{d/m}$ where $\mathbf{u}_i$ is the ith row of matrix $\mathbf{U}$, we also have:

$$\|\mathbf{Z}_{ij}\| \le \sqrt{\frac{m}{s}} \max_i \|\mathbf{u}_i\| \le C\sqrt{\frac{d}{s}} =: R.$$

Now, we apply Lemma 29 with $t = \log(d/\delta)$. For this, observe that the function $\epsilon(t)$ from (19) can be bounded as follows:

$$\begin{aligned}\epsilon(t) &= \sigma_*(\mathbf{X})t^{1/2} + R^{1/3}\sigma^{2/3}(\mathbf{X})t^{2/3} + Rt \\ &= O\bigg(\frac{d^{1/6}k^{1/3}}{s^{1/6}}\log^{2/3}(d/\delta) + \sqrt{\frac{d}{s}}\log(d/\delta)\bigg),\end{aligned}$$

so for any $k \leq d/2$, with a sufficiently large absolute constant C', if $s \geq C' \log^4(d/\delta)$ then Lemma 29 shows:

$$\Pr\left[d_H(\mathrm{spec}(\mathrm{sym}(\mathbf{SU})), \mathrm{spec}(\mathrm{sym}(\mathbf{G}))) > \sqrt{d}/6)\right] \leq \delta/2,$$

where $\mathbf{G}$ is a $k \times d$ matrix with i.i.d. standard Gaussian entries. Note that the spectrum of $\mathrm{sym}(\mathbf{G})$ consists of the k singular values of $\mathbf{G}$, their k negatives, and $d-k$ zeros, and the same is true for the spectrum of $\mathrm{sym}(\mathbf{SU})$. This follows because for any matrix $\mathbf{A}$, if $\mathbf{A} = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ is its singular value decomposition, then we can construct the eigenvectors of $\mathrm{sym}(\mathbf{A})$ associated with its positive and negative eigenvalues as follows:

$$\begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix} = \sigma_i \begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix}, \qquad \begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^\top & 0 \end{bmatrix} \begin{bmatrix} -\mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix} = -\sigma_i \begin{bmatrix} -\mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix}$$

A standard bound on the extreme singular values of a Gaussian matrix (Rudelson and Vershynin, 2009) implies that:

$$\Pr(\sqrt{d} - \sqrt{k} - t \leq \sigma_{\min}(\mathbf{G}) \leq \sigma_{\max}(\mathbf{G}) \leq \sqrt{d} + \sqrt{k} + t) \leq e^{-t^2/2}, \tag{20}$$

so when $d \geq \log(2/\delta)$ and $k \leq d/2$, we have with probability $1-\delta$ that $\sqrt{d}/2 \leq \sigma_{\min}(\mathbf{G}) \leq \sigma_{\max}(\mathbf{G}) \leq 2\sqrt{d}$. Thus, combining the Gaussian guarantee with the universality result, with probability $1-2\delta$ we have:

$$\sqrt{d}/3 \leq \sigma_{\min}(\mathbf{SU}) \leq \sigma_{\max}(\mathbf{SU}) \leq 3\sqrt{d}.$$

■

Appendix B. Linear Systems with Polynomial Spectral Decay

In this section, we discuss how our main results can be applied to derive improved time complexity of solving linear systems with polynomial spectral decay, proving Corollary 4. We also discuss and compare how algorithms and analysis from prior works apply to this setting, showing that our approach leads to a new best time complexity for a range of spectral decay profiles.

Proof of Corollary 4. Recall that we are given an $n \times n$ matrix $\mathbf{A}$ with singular values $\sigma_i = \Theta(i^{-\beta}\sigma_1)$ for $\beta > 1/2$, and our goal is to solve $\mathbf{Ax} = \mathbf{b}$. Our assumption of $\beta > 1/2$ stems only from the fact that if $0 < \beta \leq 1/2$, then such linear system can be solved in $\tilde{O}(n^2)$ using a simple stochastic gradient method, and thus the problem is trivial. According to Theorem 10, setting sketch size $k = O(n^{\frac{1}{\omega-1}})$ and $\ell = \Omega(k/\log k)$, we can solve this linear system using Algorithm 1 in time $\tilde{O}(\bar{\kappa}_\ell \bar{\kappa}_{\ell:2k} n^2)$. Thus, it suffices to bound the two condition number quantities. Observe that $\bar{\kappa}_{\ell:2k} = \tilde{O}(1)$, since σ_ℓ and σ_{2k} only differ by a factor of $O(\log^\beta(k))$, and moreover:

$$\bar{\kappa}_\ell^2 = \frac{1}{n-\ell} \sum_{i>\ell} \frac{\sigma_i^2}{\sigma_n^2} = \Theta\Big(n^{2\beta-1} \sum_{i>\ell} i^{-2\beta}\Big) = \Theta\Big(\Big(\frac{n}{\ell}\Big)^{2\beta-1}\Big).$$

Observing further that $n/\ell = \tilde{O}\big(n^{\frac{\omega-2}{\omega-1}}\big)$ concludes the proof. ■

Comparison to prior work. Next, we discuss algorithms from the most significant prior works, and how they compare in solving linear systems with polynomial spectral decay. First, consider the randomized block Kaczmarz-type solver proposed by Dereziński and Yang (2024). By choosing sketch size $k = O(n^{\frac{1}{\omega-1}})$ and $\ell = \Omega(k/\log^3 n)$, they achieve times complexity $\tilde{O}(\bar{\kappa}_\ell^2 n^2)$. Following the same derivation as in our proof of Corollary 4, this become $\tilde{O}(n^{2+\frac{\omega-2}{\omega-1}(2\beta-1)})$, which is worse than our result for any $\beta > 0.5$.

Next, we consider a preconditioning-based approach. Here, the strategy is to first construct an ℓ-rank approximation of $\mathbf{A}$ via block power iteration (Halko et al., 2011). This can be done by starting with a random Gaussian $n \times \tilde{O}(\ell)$ matrix $\mathbf{\Omega}$, then repeatedly multiplying it with $\mathbf{A}$ and $\mathbf{A}^\top$, and finally, orthogonalizing the resulting matrix $(\mathbf{A}\mathbf{A}^\top)^q \mathbf{A}\mathbf{\Omega}$ to obtain matrix $\mathbf{Q}$. Then, the matrix $\mathbf{Q}\mathbf{Q}^\top\mathbf{A}$ contains accurate approximation of the top-ℓ part of $\mathbf{A}$'s spectrum, which can be used to construct a preconditioner of the linear system. For example, Gonen et al. (2016); Musco et al. (2018b) considered an SVRG-type solver that, after being preconditioned in such a way, can solve a linear system in $\tilde{O}(\bar{\kappa}_\ell n^2)$ time. The cost of the preconditioning is dominated by the cost of the matrix multiplication, which is $\tilde{O}(n^2\ell)$ using the classical algorithm. This can be accelerated via fast rectangular matrix multiplication algorithms (Le Gall, 2012), obtaining $\tilde{O}(n^{2+\max\{0,(\omega-2)\frac{\gamma-\alpha}{1-\alpha}\}})$, where $\alpha \approx 0.32$ is the current value of the parameter of fast rectangular matrix multiplication, while $\gamma = \log_n(\ell)$. Thus, the overall cost of the procedure, combining preconditioning and solving, is $\tilde{O}(n^{2+\max\{(\omega-2)\frac{\gamma-\alpha}{1-\alpha},(1-\gamma)(\beta-0.5)\}})$, where γ can be chosen from the range $[\alpha, 1]$. After optimizing over γ, we can compare this to our Corollary 4, concluding that our guarantee is better for any $\beta \in (0.5, 1.33)$.

Positive definite matrices. We note that Corollary 4, as well as the above derivations for prior works, can be easily adapted to the setting when $\mathbf{A}$ is known to be positive definite, e.g., if it is a kernel matrix (Rasmussen and Williams, 2006). In this case, our runtime guarantee from Corollary 4 can be improved to $\tilde{O}(n^{2+\frac{\omega-2}{\omega-1}(\beta-1)/2})$ for any $\beta > 1$, and the task becomes easy for $\beta \le 1$. Here, one can show analogously as above that our results improve on the best time complexity for any $\beta \in (1, 2.66)$.

Kernel ridge regression. In the setting of kernel ridge regression, we typically introduce an explicit regularization term, resulting in the linear system $(\mathbf{A} + \lambda\mathbf{I})\mathbf{x} = \mathbf{b}$ (Erdogdu and Montanari, 2015). Depending on the value of λ, this may further reduce the computational cost of our algorithm (as well as the prior works), because it generally allows choosing a smaller value of ℓ in the algorithms. This will favor our method over the state-of-the-art, because our guarantee is the best among all approaches that use $\ell = O(n^{\frac{1}{\omega-1}})$.

References

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

Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. *Advances in Neural Information Processing Systems*, 28, 2015.

Seth J Alderman, Roan W Luikart, and Nicholas F Marshall. Randomized Kaczmarz with geometrically smoothed momentum. *arXiv preprint arXiv:2401.09415*, 2024.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18(221):1–51, 2018.

Owe Axelsson and Gunhild Lindskog. On the rate of convergence of the preconditioned conjugate gradient method. *Numerische Mathematik*, 48:499–523, 1986.

Raghu Bollapragada, Tyler Chen, and Rachel Ward. On the fast convergence of minibatch heavy ball momentum. *IMA Journal of Numerical Analysis*, page drae033, 2024.

Tatiana Brailovskaya and Ramon van Handel. Universality and sharp matrix concentration inequalities. *Geometric and Functional Analysis*, pages 1–105, 2024.

Mark Braverman, Elad Hazan, Max Simchowitz, and Blake Woodworth. The gradient complexity of linear regression. In *Conference on Learning Theory*, pages 627–647. PMLR, 2020.

David Burt, Carl Edward Rasmussen, and Mark Van Der Wilk. Rates of convergence for sparse variational Gaussian process regression. In *International Conference on Machine Learning*, pages 862–871. PMLR, 2019.

T Tony Cai, Zongming Ma, and Yihong Wu. Sparse PCA: Optimal rates and adaptive estimation. *The Annals of Statistics*, 41(6):3074–3110, 2013.

Mireille Capitaine, Catherine Donati-Martin, and Delphine Féral. The largest eigenvalues of finite rank deformation of large wigner matrices: Convergence and nonuniversality of the fluctuations. *The Annals of Probability*, pages 1–47, 2009.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004. doi: https://doi.org/10.1016/S0304-3975(03)00400-6.

Shabarish Chenakkod, Michał Dereziński, Xiaoyu Dong, and Mark Rudelson. Optimal embedding dimension for sparse subspace embeddings. In *56th Annual ACM Symposium on Theory of Computing*, 2024.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 163–172, 2015.

Michael B Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1758–1777. SIAM, 2017.

Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 1–6, 1987.

James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. *Numerische Mathematik*, 108(1):59–91, 2007.

Michał Dereziński and Michael W Mahoney. Determinantal point processes in randomized numerical linear algebra. *Notices of the American Mathematical Society*, 68(1):34–45, 2021.

Michał Dereziński and Michael W Mahoney. 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*, pages 6470–6479, 2024.

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

Michał Dereziński and Jiaming Yang. Solving dense linear systems faster than via preconditioning. In *56th Annual ACM Symposium on Theory of Computing*, 2024.

Michał Dereziński, Kenneth L Clarkson, Michael W Mahoney, and Manfred K Warmuth. Minimax experimental design: Bridging the gap between statistical and worst-case approaches to least squares regression. In *Conference on Learning Theory*, pages 1050–1069. PMLR, 2019.

Michał Dereziński, Rajiv Khanna, and Michael W Mahoney. Improved guarantees and a multiple-descent curve for column subset selection and the Nyström method. *Advances in Neural Information Processing Systems*, 33:4953–4964, 2020.

Michał Dereziński, Jonathan Lacotte, Mert Pilanci, and Michael W Mahoney. Newton-LESS: Sparsification without trade-offs for the sketched Newton update. In *Advances in Neural Information Processing Systems*, volume 34, pages 2835–2847. Curran Associates, Inc., 2021.

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

Aymeric Dieuleveut, Nicolas Flammarion, and Francis Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *Journal of Machine Learning Research*, 18 (101):1–51, 2017.

Petros Drineas and Michael W Mahoney. RandNLA: randomized numerical linear algebra. *Communications of the ACM*, 59(6):80–90, 2016.

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

Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

Nicole Eikmeier and David F Gleich. Revisiting power-law distributions in spectra of real world networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 817–826, 2017.

Murat A Erdogdu and Andrea Montanari. Convergence rates of sub-sampled newton methods. *Advances in Neural Information Processing Systems*, 28, 2015.

Mathieu Even, Raphaël Berthier, Francis Bach, Nicolas Flammarion, Pierre Gaillard, Hadrien Hendrikx, Laurent Massoulié, and Adrien Taylor. A continuized view on Nesterov acceleration for stochastic gradient descent and randomized gossip. *arXiv preprint arXiv:2106.07644*, 2021.

Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *International Conference on Machine Learning*, pages 2540–2548. PMLR, 2015.

Nidham Gazagnadou, Mark Ibrahim, and Robert M Gower. Ridgesketch: A fast sketching based solver for large scale ridge regression. *SIAM Journal on Matrix Analysis and Applications*, 43(3):1440–1468, 2022.

Gene H Golub and Richard S Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. *Numerische Mathematik*, 3(1):157–168, 1961.

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

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

Robert Gower, Dmitry Kovalev, Felix Lieder, and Peter Richtárik. RSN: randomized subspace newton. *Advances in Neural Information Processing Systems*, 32, 2019.

Robert M Gower and Peter Richtárik. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1660–1690, 2015a.

Robert M Gower and Peter Richtárik. Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1380–1409, 2017.

Robert Mansel Gower and Peter Richtárik. Stochastic dual ascent for solving linear systems. *arXiv preprint arXiv:1512.06890*, 2015b.

Walid Hachem, Philippe Loubaton, and Jamal Najim. Deterministic equivalents for certain functionals of large random matrices. *The Annals of Applied Probability*, 17(3):875 – 930, 2007. doi: 10.1214/105051606000000925.

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

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

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

Magnus Rudolph Hestenes, Eduard Stiefel, et al. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.

J Ben Hough, Manjunath Krishnapur, Yuval Peres, and Bálint Virág. Determinantal processes and independence. *Probability surveys*, 3:206–229, 2006.

G.N. Hounsfield. Computerized transverse axial scanning (tomography): Part I. Description of the system. *British J. Radiol.*, 46:1016–1022, 1973.

Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of Statistics*, 29(2):295–327, 2001.

S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bull. Int. Acad. Polon. Sci. Lett. Ser. A*, pages 335–357, 1937.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Francois Le Gall. Faster algorithms for rectangular matrix multiplication. In *2012 IEEE 53rd annual symposium on foundations of computer science*, pages 514–523. IEEE, 2012.

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

Daniel LeJeune, Pratik Patil, Hamid Javadi, Richard G Baraniuk, and Ryan J Tibshirani. Asymptotics of the sketched pseudoinverse. *SIAM Journal on Mathematics of Data Science*, 6(1):199–225, 2024.

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

Jörg Liesen and Zdenek Strakos. *Krylov subspace methods: principles and analysis.* Numerical Mathematics and Scie, 2013.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. *Advances in neural information processing systems*, 28, 2015.

Ji Liu and Stephen Wright. An accelerated randomized Kaczmarz algorithm. *Mathematics of Computation*, 85(297):153–178, 2016.

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

Xiangrui Meng, Michael A Saunders, and Michael W Mahoney. LSRN: A parallel iterative solver for strongly over-or underdetermined systems. *SIAM Journal on Scientific Computing*, 36(2):C95–C118, 2014.

Cameron Musco, Christopher Musco, and Aaron Sidford. Stability of the Lanczos method for matrix function approximation. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1605–1624. SIAM, 2018a.

Cameron Musco, Praneeth Netrapalli, Aaron Sidford, Shashanka Ubaru, and David P Woodruff. Spectrum approximation beyond fast matrix multiplication: Algorithms and hardness. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2018b.

Mojmir Mutny, Michał Dereziński, and Andreas Krause. Convergence analysis of block coordinate algorithms with determinantal sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 3110–3120. PMLR, 2020.

Frank Natterer. *The mathematics of computerized tomography*. SIAM, 2001.

Ion Necoara, Peter Richtárik, and Andrei Patrascu. Randomized projection methods for convex feasibility: Conditioning and convergence rates. *SIAM Journal on Optimization*, 29(4):2814–2852, 2019.

Jelani Nelson and Huy L Nguyên. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *2013 IEEE 54th annual Symposium on Foundations of Computer Science*, pages 117–126. IEEE, 2013.

AS Nemirovsky and DB Yudin. *Problem complexity and method efficiency in optimization*. John Wiley & Sons, Inc New York, 1983.

Y. E. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Doklady Akademii Nauk SSSR*, 269(3):543, 1983.

Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Samet Oymak and Joel A Tropp. Universality laws for randomized dimension reduction, with applications. *Information and Inference: A Journal of the IMA*, 7(3):337–446, 2018.

Victor Pan. *How to multiply matrices faster*. Springer-Verlag, 1984.

Richard Peng and Santosh Vempala. Solving sparse linear systems faster than matrix multiplication. In *Proceedings of the 2021 ACM-SIAM symposium on discrete algorithms (SODA)*, pages 504–521. SIAM, 2021.

Amelia Perry, Alexander S Wein, Afonso S Bandeira, and Ankur Moitra. Optimality and sub-optimality of pca i: Spiked random matrix models. *The Annals of Statistics*, 46(5): 2416–2451, 2018.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

Elizaveta Rebrova and Deanna Needell. On block Gaussian sketching for the Kaczmarz method. *Numerical Algorithms*, 86:443–473, 2021.

Peter Richtárik and Martin Takác. Stochastic reformulations of linear systems: algorithms and convergence theory. *SIAM Journal on Matrix Analysis and Applications*, 41(2):487–524, 2020.

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

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

Francisco Rubio and Xavier Mestre. Spectral convergence for a general class of random matrices. *Statistics and Probability Letters*, 81(5):592–602, 2011. doi: https://doi.org/10.1016/j.spl.2011.01.004.

Mark Rudelson and Roman Vershynin. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics*, 62(12):1707–1739, 2009. doi: https://doi.org/10.1002/cpa.20294.

Yousef Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of computation*, 37(155):105–126, 1981.

Huaiyu Zhu Santa, Huaiyu Zhu, Christopher K. I. Williams, Richard Rohwer, and Michal Morciniec. Gaussian regression and optimal finite dimensional linear models. In *Neural Networks and Machine Learning*, pages 167–184. Springer-Verlag, 1997.

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pages 143–152. IEEE, 2006.

Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, page 166–179, New York, NY, USA, 2001. Association for Computing Machinery.

J.W. Silverstein and Z.D. Bai. On the empirical distribution of eigenvalues of a class of large dimensional random matrices. *Journal of Multivariate Analysis*, 54(2):175–192, 1995. doi: https://doi.org/10.1006/jmva.1995.1051.

Daniel A Spielman and Jaeoh Woo. A note on preconditioning by low-stretch spanning trees. *arXiv preprint arXiv:0903.2816*, 2009.

Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4): 354–356, Aug 1969. ISSN 0945-3245.

T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.*, 15(2):262–278, 2009.

Ling Tang, Yajie Yu, Yanjun Zhang, and Hanyu Li. Sketch-and-project methods for tensor linear systems. *Numerical Linear Algebra with Applications*, 30(2):e2470, 2023.

Joel A Tropp. Improved analysis of the subsampled randomized Hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126, 2011.

Ramon Van Handel. Probability in high dimension. *Lecture Notes (Princeton University)*, 2014.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018. doi: 10.1017/9781108231596.

Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 887–898, 2012.

Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3792–3835, 2024. doi: 10.1137/1.9781611977912.134.

Carsten H Wolters, Harald Köstler, Christian Möller, Jochen Härdtlein, Lars Grasedyck, and Wolfgang Hackbusch. Numerical mathematics of the subtraction method for the modeling of a current dipole in EEG source reconstruction using finite element head models. *SIAM Journal on Scientific Computing*, 30(1):24–45, 2008.

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

Jianlin Xia, Shivkumar Chandrasekaran, Ming Gu, and Xiaoye S Li. Superfast multifrontal method for large structured linear systems of equations. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1382–1411, 2010.

Peng Xu, Fred Roosta, and Michael W Mahoney. Newton-type methods for non-convex optimization under inexact hessian information. *Mathematical Programming*, 184(1):35–70, 2020.

Haishan Ye, Luo Luo, and Zhihua Zhang. Nesterov's acceleration for approximate newton. *Journal of Machine Learning Research*, 21(142):1–37, 2020.

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