Faster Linear Systems and Matrix Norm Approximation via Multi-level Sketched Preconditioning

Michał Dereziński* Christopher Musco† Jiaming Yang‡

Abstract

We present a new class of preconditioned iterative methods for solving linear systems of the form $\mathbf{Ax} = \mathbf{b}$. Our methods are based on constructing a low-rank Nyström approximation to $\mathbf{A}$ using sparse random matrix sketching. This approximation is used to construct a preconditioner, which itself is inverted quickly using additional levels of random sketching and preconditioning.

We prove that the convergence of our methods depends on a natural *average condition number* of $\mathbf{A}$, which improves as the rank of the Nyström approximation increases. Concretely, this allows us to obtain faster runtimes for a number of fundamental linear algebraic problems:

1. We show how to solve any $n \times n$ linear system that is well-conditioned except for k outlying large singular values in $\tilde{O}(n^{2.065} + k^{\omega})$ time, improving on a recent result of [Dereziński, Yang, STOC 2024] for all $k \gtrsim n^{0.78}$.
2. We give the first $\tilde{O}(n^2 + d_\lambda{}^{\omega})$ time algorithm for solving a regularized linear system $(\mathbf{A} + \lambda\mathbf{I})\mathbf{x} = \mathbf{b}$, where $\mathbf{A}$ is positive semidefinite with effective dimension $d_\lambda = \mathrm{tr}(\mathbf{A}(\mathbf{A} + \lambda\mathbf{I})^{-1})$. This problem arises in applications like Gaussian process regression.
3. We give faster algorithms for approximating Schatten p-norms and other matrix norms. For example, for the Schatten 1-norm (nuclear norm), we give an algorithm that runs in $\tilde{O}(n^{2.11})$ time, improving on an $\tilde{O}(n^{2.18})$ method of [Musco et al., ITCS 2018].

All results are proven in the real RAM model of computation. Interestingly, previous state-of-the-art algorithms for most of the problems above relied on stochastic iterative methods, like stochastic coordinate and gradient descent. Our work takes a completely different approach, instead leveraging tools from matrix sketching.

1 Introduction

We consider the complexity of solving a system of linear equations: given an $n \times n$ matrix $\mathbf{A}$ and an n-dimensional vector $\mathbf{b}$, find $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{b}$. This ubiquitous task in numerical linear algebra has applications across data science and machine learning, the physical sciences, engineering, and more. Despite many existing methods for solving linear systems, they remain a major computational bottleneck, so faster algorithms are a subject of active research. Much of the progress on faster algorithms concentrates on designing methods tailored for matrices with special structure, such as Laplacian, Toeplitz, or Hankel matrices, among others [KKM79, XXG12, ST14a, KMP12, KS16, CKK+18, PV21]. Progress on general, unstructured linear systems has been slower.

One direction for developing faster algorithms is to improve on fast matrix multiplication. Thanks to Strassen's reduction showing that matrix inversion is equivalent to matrix multiplication [Str69, Pan84, CW87, Wil12], general linear systems can be solved in $O(n^{\omega})$ time, where $\omega < 2.372$ is the current best known matrix multiplication exponent [WXXZ23]. Another approach is to solve linear systems via iterative refinement, for instance using deterministic methods such as the Conjugate Gradient (CG) or Lanczos algorithms [HS52], or stochastic approaches like randomized coordinate descent [SV09, LL10, LS13]. Iterative methods tend to be faster in many practical settings, as their runtime typically scales only quadratically with n, i.e., as $O(n^2)$. However, the runtime of typical iterative methods involves a multiplicative factor depending on the condition number of $\mathbf{A}$ (the ratio between its largest and smallest singular values) or related parameters, making them largely incomparable to solvers based on fast matrix multiplication.

*University of Michigan (derezin@umich.edu)
†New York University (cmusco@nyu.edu)
‡University of Michigan (jiamyang@umich.edu)

One way to address this issue is to introduce a problem parameter k, such that any ill-conditioned part of $\mathbf{A}$ is restricted to a k-dimensional subspace. Formally, we ask:

PROBLEM 1.1. *What is the time complexity of solving an $n \times n$ linear system $\mathbf{Ax} = \mathbf{b}$ such that matrix $\mathbf{A}$ has at most k singular values larger than $O(1)$ times its smallest singular value?*

When $k = n$, our fastest methods for solving Problem 1.1 run in $O(n^\omega)$ time via fast matrix multiplication. When $k = 0$, the problem can be solved in $\tilde{O}(n^2)$ time via any standard iterative solver such as CG. We are interested in the arguably more interesting intermediate regime, where $\mathbf{A}$ is a combination of a low-rank ill-conditioned matrix and a full-rank well-conditioned one. Such matrices arise often in practice, either due to various forms of regularization that are prevalent in machine learning, statistics and optimization [BV04,ZDW13,DW18,DM24], or due to the presence of isotropic noise coming from measurement error, rounding, or compression [LW11,LGW+21].

A standard approach to Problem 1.1 is to construct an approximation to the rank k subspace identified by $\mathbf{A}$'s top singular values. That information can be used to precondition an iterative solver so that it runs in $\tilde{O}(n^2)$ time [HMT11]. Constructing a sufficiently accurate approximation to the top subspace requires (block) power iteration or related methods [MM15], which take $\tilde{O}(n^{\omega(1,1,\log_n k)})$ time, where $\omega(1, 1, \log_n k) \in [2, \omega]$ is the exponent of rectangular matrix multiplication between an $n \times n$ and $n \times k$ matrix [LG12].[1] This approach already interpolates between the two extremes of the problem. In fact, it achieves a near-optimal $\tilde{O}(n^2)$ runtime for $k = O(n^{0.32})$.

Recently, [DY24] give the first improvement on this baseline by presenting a randomized Kaczmarz-like iterative method that solves Problem 1.1 in time $\tilde{O}(n^2 + nk^{\omega-1})$. This result leads to an optimal $\tilde{O}(n^2)$ runtime for any $k = O(n^{\frac{1}{\omega-1}}) = O(n^{0.73})$, and for larger k it gradually degrades to $O(n^\omega)$.

1.1 Main Results It might appear as though the result of [DY24] is optimal. In particular, the currently best known algorithms for finding even a coarse (Frobenius norm error) approximation of the top rank k subspace of $\mathbf{A}$ also require $\tilde{O}(n^2 + nk^{\omega-1})$ time [CW13, CEM+15, CMM17, CCKW22], and finding such an approximation seems like a prerequiste for solving Problem 1.1. The contribution of this paper is to present an algorithm that breaks through this complexity barrier. Along the way, we give new state-of-the-art runtimes for two other central problems in linear algebra that are closely related to Problem 1.1: kernel ridge regression [ACW17] and Schatten norm approximation [MNS+18].

Concretely, we prove the following result, which improves on the previously best known time complexity of $\tilde{O}(n^2 + nk^{\omega-1})$ for Problem 1.1 for all $k = \Omega(n^{0.78})$. See Figure 1 for a comparison.

THEOREM 1.1. (MAIN RESULT, INFORMAL THEOREM 3.1) *Given an invertible $n \times n$ matrix $\mathbf{A}$ with at most k singular values larger than $O(1)$ times its smallest singular value, and a length n vector $\mathbf{b}$, there is an algorithm that, with high probability, computes $\tilde{\mathbf{x}}$ such that $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\| \leq \epsilon\|\mathbf{b}\|$ in time[2]:*

$$\tilde{O}\left(\left(n^{2.065} + k^\omega\right) \cdot \log^3 1/\epsilon\right).$$

To understand the significance of Theorem 1.1, consider the following special instance of Problem 1.1, where $\mathbf{A}$ is a block-diagonal matrix with one $k \times k$ block that contains an arbitrary ill-conditioned matrix, and a second $(n-k) \times (n-k)$ block that contains a well-conditioned (but non-trivial) matrix. Solving such a linear system is equivalent to solving a small worst-case $k \times k$ linear system and a dense, well-condition system. Thus, under the assumption that the worst-case time complexity of linear systems is governed by the complexity of matrix multiplication, we can lower bound the cost of solving Problem 1.1 with current fast matrix multiplication by $\Omega(n^2 + k^\omega)$ (this is formalized in Theorem 7.1). In comparison, our approach yields $\tilde{O}(n^{2.065} + k^\omega)$ time, which matches the lower bound up to $n^{0.065}$ everywhere and up to logarithmic factors for $k = \Omega(n^{0.87})$.

Surprisingly, our algorithm departs from the stochastic optimization approach used in the prior work of [DY24]. Instead, we employ a deterministic solver with a randomized preconditioner. To do so, we combine two main ingredients. First, building on prior work [FTU23, DEF+23], we show that even a very coarse approximation to $\mathbf{A}$'s top rank k subspace can be used to construct a "good enough" preconditioner. In particular, we can replace the use of algorithms like block power method with more efficient, but less accurate, linear-time sketching methods

[1] This corresponds to the $O(n^2 k)$ time for classical multiplication of $n \times n$ and $n \times k$ matrices.

[2] We use $\tilde{O}(\cdot)$ to hide logarithmic dependencies on the dimension n and the condition number of $\mathbf{A}$.

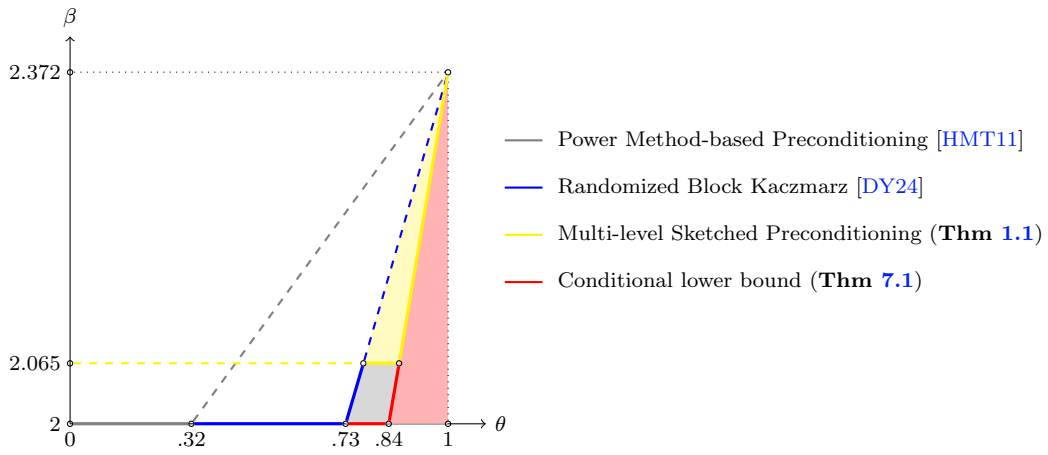

Figure 1: Time complexity for solving an $n \times n$ linear system with $k = n^\theta$ large singular values under current matrix multiplication exponent $\omega \approx 2.372$. The x-axis denotes the exponent θ, while the y-axis denotes the exponent β in the time complexity $\tilde{O}(n^\beta)$. The yellow line is our work (Theorem 1.1), with the yellow area showing the complexity improvement compared to prior work. The red line denotes a lower bound for the problem, which we prove in Theorem 7.1. The red area is unachievable under the assumption that solving general dense linear systems requires $\Omega(n^\omega)$ time.

for low-rank approximation [CW13]. Second, we show that it is possible to avoid the final step of such methods, which requires a prohibitive $\tilde{O}(nk^{\omega-1})$ cost to explicitly construct a rank k subspace approximating $\mathbf{A}$'s top singular vectors. Instead, using $\tilde{O}(nk + k^\omega)$ preprocessing time, we show how to maintain a data structure that lets us perform inexact matrix-vector products with an inverted low-rank preconditioner. The data structure involves additional levels of solving linear systems with randomized sketching and preconditioning methods, resulting in a framework that we call **Multi-level Sketched Preconditioning (MSP)**. We discuss details of our approach further in Section 1.2. Before doing so, we highlight two additional applications of the framework beyond Theorem 1.1.

Regularized linear systems. An important case of linear systems that are well-conditioned except for a few large singular values are those that are explicitly regularized by adding a scaled identity, $\lambda\mathbf{I}$, to a positive semidefinite matrix $\mathbf{A}$. This setting arises, for instance, in kernel ridge regression [EAM14, RCR17, MM17, ACW17, FTU23] and second-order optimization [Lev44, Mar63, MS12, JJM23]. Prior results have shown that such linear systems can be solved in $\tilde{O}(n^2 + nd_\lambda{}^{\omega-1})$ time, where $d_\lambda = \mathrm{tr}(\mathbf{A}(\mathbf{A} + \lambda\mathbf{I})^{-1}) \leq n$ is the so-called λ-effective dimension of the problem, often much smaller than n [ACW17, MM17]. We improve on this by using Multi-level Sketched Preconditioning, leading to a time complexity of $\tilde{O}(n^2 + d_\lambda{}^\omega)$, which is optimal (up to log factors).

Theorem 1.2. (Regularized linear systems, informal Theorem 3.2) *Given an $n \times n$ positive semidefinite matrix $\mathbf{A}$, an n-dimensional vector $\mathbf{b}$, and $\lambda > 0$, there is an algorithm that, with high probability, computes $\tilde{\mathbf{x}}$ such that $\|(\mathbf{A} + \lambda\mathbf{I})\tilde{\mathbf{x}} - \mathbf{b}\| \leq \epsilon\|\mathbf{b}\|$ in time:*

$$\tilde{O}\left(\left(n^2 + d_\lambda{}^\omega\right) \cdot \log^3 1/\epsilon\right), \quad \textit{where} \quad d_\lambda = \mathrm{tr}(\mathbf{A}(\mathbf{A} + \lambda\mathbf{I})^{-1}).$$

Matrix norm estimation. Linear systems often arise in algorithms for solving other matrix problems, including linear and semidefinite programming [CLS21, JKL+20], least squares and l_p regression [Sar06, RT08, MM13, CP15, CD21], and for estimating various matrix properties [MNS+18]. Our new algorithms can be used to speed up any of these tasks. As a motivating example, we show how to improve methods for estimating various matrix norms, a fundamental problem in linear algebra. We build on an approach of [MNS+18], which uses regularized linear system solves to build rational function approximations that, when combined with stochastic trace estimation methods [Hut90], allow for the estimation of various functions of the singular values faster than

$O(n^\omega)$ time, i.e. faster than the time it takes to compute all singular values outright.[3]

Perhaps the most important example is the sum of the singular values, i.e., the Schatten 1-norm $\|\mathbf{A}\|_1$ (a.k.a. nuclear norm). [MNS+18] shows how to obtain a $(1+\epsilon)$ multiplicative approximation to $\|\mathbf{A}\|_1$ in time $\tilde{O}(n^{2.18}\text{poly}(1/\epsilon))$. This is a surprising result, as no previous methods were able to beat the $O(n^\omega)$ complexity of computing a full SVD. Moreover, work in fine-grained complexity (specifically, on triangle detection lower bounds) suggests that $O(n^\omega)$ is tight for high-accuracy approximation (i.e., $\epsilon = \text{poly}(1/n)$) [MNS+18, WW10]. By replacing the stochastic iterative methods used in [MNS+18] with our MSP solvers, we make further progress on this problem.

THEOREM 1.3. (SCHATTEN 1-NORM ESTIMATION, COROLLARY OF THEOREM 3.3) *Given an $n\times n$ matrix $\mathbf{A}$ and $\epsilon > 0$, there is an algorithm that, with high probability, computes $X \in (1\pm\epsilon)\|\mathbf{A}\|_1$ in time:*

$$\tilde{O}\big(n^{2.11}\text{poly}(1/\epsilon)\big).$$

In fact, we show that our linear solver improves the exponent of n in the time complexity of Schatten p-norm estimation for any $p \in (0, 1.5)$, and it can also be used to approximate other matrix norms such as the Ky Fan and Orlicz norms. See Section 3.2 for a discussion.

1.2 Our Techniques The starting point for our Multi-level Sketched Preconditioning comes from work on solving regularized systems of the form $(\mathbf{A}+\lambda\mathbf{I})\mathbf{x} = \mathbf{b}$, where $\mathbf{A}$ is positive definite (PD). Here, a central technique for preconditioning is the Nyström method [Nys30, WS01, GM16, MM17], which uses randomized sketching or subsampling to build a low-rank approximation for $\mathbf{A}$. As an illustration, let $\mathbf{S} \in \mathbb{R}^{l\times n}$ be a subsampling matrix, so that $\mathbf{AS}^\top$ contains a random subset of l columns from $\mathbf{A}$. The classical Nyström approximation is defined as $\hat{\mathbf{A}}_{\text{nys}} = \mathbf{CW}^{-1}\mathbf{C}^\top$, where $\mathbf{C} = \mathbf{AS}^\top$ is the column submatrix and $\mathbf{W} = \mathbf{SAS}^\top$ is the corresponding $l\times l$ principal submatrix of $\mathbf{A}$. This can be extended to other sketching methods by replacing $\mathbf{S}$ with a sparse random matrix (CountSketch, OSNAP, LESS, etc. [CW13, NN13, Coh16, DLDM21]) or a Subsampled Randomized Hadamard Transform (SRHT, [AC09, Tro11]), all of which can be multiplied by $\mathbf{A}$ in $\tilde{O}(n^2)$ time. With sufficiently large sketch size (namely, proportional to the effective dimension $d_\lambda = \text{tr}(\mathbf{A}(\mathbf{A}+\lambda\mathbf{I})^{-1})$), one can show that $\mathbf{M} = \hat{\mathbf{A}}_{\text{nys}} + \lambda\mathbf{I}$ is a good preconditioner for $\mathbf{A}+\lambda\mathbf{I}$, i.e., the condition number of $\mathbf{M}^{-1}(\mathbf{A}+\lambda\mathbf{I})$ is constant. At the same time, $\mathbf{M}^{-1}$ can be applied quickly due to its decomposition into $\mathbf{C}$ and $\mathbf{W}$. Concretely, using the inversion formula $\mathbf{M}^{-1} = \frac{1}{\lambda}\big(\mathbf{I} - \mathbf{C}(\mathbf{C}^\top\mathbf{C}+\lambda\mathbf{W})^{-1}\mathbf{C}^\top\big)$, we can compute $\mathbf{M}^{-1}\mathbf{r}$ for any vector $\mathbf{r}$ in $\tilde{O}(nl)$ time after precomputing $(\mathbf{C}^\top\mathbf{C}+\lambda\mathbf{W})^{-1}$ in $\tilde{O}(nl^{\omega-1})$ time.

Average Condition Number Bounds via Nyström Preconditioning. We improve on standard Nyström preconditioning in two important ways. First, replacing the subsampling matrix $\mathbf{S}$ with a sparse sketching matrix [CNW16], we leverage a moment version of the oblivious subspace embedding property in conjunction with a spectral norm error low-rank approximation bound, to show that even when solving an *unregularized* system $\mathbf{Ax} = \mathbf{b}$, a preconditioner of the form $\mathbf{M} = \hat{\mathbf{A}}_{\text{nys}} + \lambda\mathbf{I}$ can significantly improve conditioning. Note that the scalar λ is now a parameter of the algorithm, and not of the problem, as was the case for standard Nyström preconditioning. We prove that, using Nyström approximation with sketch size $\tilde{O}(l)$ for any integer l, we can choose λ so that the condition number of the preconditioned system $\mathbf{M}^{-1}\mathbf{A}$ can be reduced to $\tilde{O}(\frac{n}{l}\cdot\bar{\kappa}_l)$, where $\bar{\kappa}_l = \frac{1}{n-l}\sum_{i>l}\frac{\sigma_i}{\sigma_n}$ is the average condition number of $\mathbf{A}$, excluding its top l singular values (here, σ_i are the singular values of $\mathbf{A}$ listed in a decreasing order). We observe that the quality of the preconditioner exhibits a dependence on l both through the $\frac{n}{l}$ factor and the condition number $\bar{\kappa}_l$. Thus, obtaining the best condition number requires carefully tuning the parameter ℓ for each application. For example, if $\mathbf{A}$ has at most k large singular values, as in Problem 1.1, then $\bar{\kappa}_l = O(1)$ for any $l \geq k$, but due to the dependence on $\frac{n}{l}$, increasing l past k still improves the preconditioner. Unfortunately, this happens at the expense of increasing the $\tilde{O}(nl^{\omega-1})$ cost of inverting the Nyström approximation, which motivates our second contribution.

Two-level Preconditioning for Positive Definite Systems. Instead of computing $\mathbf{M}^{-1} = \frac{1}{\lambda}\big(\mathbf{I} - \mathbf{C}(\mathbf{C}^\top\mathbf{C}+\lambda\mathbf{W})^{-1}\mathbf{C}^\top\big)$ *explicitly*, as in standard Nyström preconditioning, we focus on simply implementing matrix-vector

[3]Concretely, [MNS+18] estimates quantities of the form $\text{tr}(f(\mathbf{A}))$, with the nuclear norm equal to the case where $f(x) = |x|$. Stochastic trace estimation methods like Hutchinson's estimator and related techniques approximate $\text{tr}(f(\mathbf{A}))$ by computing $\mathbf{g}^T f(\mathbf{A})\mathbf{g}$ for randomly chosen $\mathbf{g}$ [MMMW21].

multiplications with $\mathbf{M}^{-1}$, which suffice to apply a preconditioned iterative method like CG or Lanczos. The main challenge in doing so is to implement efficient multiplications with $(\mathbf{C}^\top\mathbf{C}+\lambda\mathbf{W})^{-1}$, i.e., to solve the system $(\mathbf{C}^\top\mathbf{C}+\lambda\mathbf{W})\mathbf{x}=\mathbf{r}$. We show how to do this using a *second level of sketching and preconditioning*. In particular, since $\mathbf{C}$ is a tall $n\times l$ matrix, the linear system $(\mathbf{C}^\top\mathbf{C}+\lambda\mathbf{W})\mathbf{x}=\mathbf{r}$ can be solved efficiently by preconditioning with $\mathbf{C}^\top\mathbf{\Phi}^\top\mathbf{\Phi}\mathbf{C}+\lambda\mathbf{W}$, where $\mathbf{\Phi}$ is an $O(l)\times n$ sketching matrix (specifically, an oblivious subspace embedding). The preconditioner $\mathbf{C}^\top\mathbf{\Phi}^\top\mathbf{\Phi}\mathbf{C}+\lambda\mathbf{W}$ can be constructed and inverted in just $\tilde{O}(nl+l^\omega)$ time, which leads to an upfront cost of $\tilde{O}(nl+l^\omega)$ for implementing $\tilde{O}(nl)$ time matrix-vector multiplications with $\mathbf{M}^{-1}$. This is a significant improvement on the $\tilde{O}(nl^{\omega-1})$ required by explicit inversion. Overall, we obtain a complexity of $\tilde{O}(n^2\sqrt{n/l}+l^\omega)$ for solving Problem 1.1 when $\mathbf{A}$ is positive definite. Optimizing over the parameter l gives a final runtime bound of $\tilde{O}(n^{2.065}+k^\omega)$ time for positive definite $\mathbf{A}$, which is our first step towards proving the general version of Theorem 3.2, which does not assume positive definiteness. See Section 4, Theorem 4.1, for details.

We note that to analyze the approach above, we leverage a stability analysis of the preconditioned Lanczos method, which ensures that an optimal convergence rate is still obtained even when the preconditioner $\mathbf{M}^{-1}$ is applied inexactly. We give this analysis in Section 6, building on prior work on the stability of the unpreconditioned Lanczos method [Pai71, Pai76, Gre89, MMS18]. Many alternative approaches would also suffice. For example, we could have obtained the same running times by using preconditioned Chebyshev iteration in place of Lanczos. The robustness of Chebyshev iteration to inexact applications of $\mathbf{M}^{-1}$ is well understood, and leveraged, e.g., in fast solvers for Laplacian systems [ST14b]. However, the method tends to converge much slower in practice than more popular methods like Lanczos or the Conjugate Gradient method. Thus, providing a stability analysis of the preconditioned Lanczos method is of independent interest beyond its application in our algorithms. Broadly, an increasing number of algorithms in numerical linear algebra combine iterative methods with inexact "inner loops", often applied using randomized techniques. This approach has found applications in spectral density estimation [BKM22], quantum inspired linear algebra [BT24], and a number of other problems [OSV12]. All of this work hinges on stability analysis akin to our results on the preconditioned Lanczos method.

Three-level Preconditioning for Indefinite Systems. The above strategy, which involves two levels of preconditioning, is restricted to positive definite matrices. Extending the approach to arbitrary linear systems requires additional work. A natural first attempt is to reduce an arbitrary $\mathbf{A}\mathbf{x}=\mathbf{b}$ linear system to a positive (semi-)definite linear system via the normal equations, $\mathbf{A}^\top\mathbf{A}\mathbf{x}=\mathbf{A}^\top\mathbf{b}$. However, if we apply the Nyström preconditioning approach discussed above, we realize that we require matrices $\mathbf{C}=\mathbf{A}^\top\mathbf{A}\mathbf{S}^\top$ and $\mathbf{W}=\mathbf{S}\mathbf{A}^\top\mathbf{A}\mathbf{S}^\top$, which can no longer be computed in $\tilde{O}(n^2)$ time, even if $\mathbf{S}$ is a sparse random matrix. To address this, we show that the matrix $(\mathbf{C}^\top\mathbf{C}+\lambda\mathbf{W})^{-1}$ can nevertheless be applied efficiently by preconditioning with the approximation $(\mathbf{W}^2+\lambda\mathbf{W})^{-1}=\frac{1}{\lambda}\big(\mathbf{W}^{-1}-(\mathbf{W}+\lambda\mathbf{I})^{-1}\big)$. This preconditioner, in turn, can be applied by solving two linear systems associated with $\mathbf{W}^{-1}$ and $(\mathbf{W}+\lambda\mathbf{I})^{-1}$, both using a *third level* of sketching and preconditioning. Details are included in Section 5.

While Theorem 1.1 focuses on solving linear systems with k large singular values, our main technical result (Theorem 3.1) shows that Multi-level Sketched Preconditioning can be used for any linear system, with the caveat that the condition number $\bar{\kappa}_l$ will appear in the bounds. This more general setting is used to obtain our improved Schatten norm algorithms, which are based on the work of [MNS+18], and require solving linear systems where $\bar{\kappa}_l$ can be bounded by $O(\sqrt{n/l})$. Optimizing over l yields the final time complexity of $\tilde{O}(n^{2.11})$. See Section 3.2 for details.

1.3 Additional Related Work

There has been significant prior work on solving linear systems with a small number of outlying singular values, or more generally, whose condition number can be reduced by eliminating large singular values. As discussed, our work is most closely related to methods that use coarse low-rank approximations to build preconditioners for such linear systems, an approach that has been extremely popular for solving large regularized regression problems arising e.g., in Gaussian process regression (kernel regression) problems [RCR17, MB17, MCRR20, FTU23, DEF+23].

As discussed in the previous section, techniques like Fast Randomized Hadamard Transform, sparse sketching matrices, or leverage score sampling [AM15, CMM17] can be used to construct a low-rank approximation in just $\tilde{O}(n^2)$ time. When $\mathbf{A}$ is PSD, subquadratic time is even possible [MW17]. This is in contrast to ℓ-rank approximation methods based on power iteration or other Krylov methods, which require $\tilde{O}(n^2\ell)$ time [GOSS16]. However, previous methods for actually applying the low-rank preconditioner required at least $O(n\ell^{\omega-1})$,

motivating our Multi-level Sketched Preconditioning, which avoids the need for explicitly orthogonalizing an $n \times \ell$ matrix. The goal of avoiding orthogonalization also arises in other problems related to low-rank approximation, notably the *principal component regression (PCR)* problem [FMMS16, JS19]. Our MSP methods may be able to improve PCR algorithms, which rely on black-box solvers for regularized regression.

Another related line of work seeks to understand how iterative methods like the conjugate gradient method behave for a matrix with few outlying singular values [SW09]. In fact, for Problem 1.1, it is well known that the *unpreconditioned* CG or Lanczos methods will converge in roughly $k + \kappa$ steps, where κ is the condition number of $\mathbf{A}$'s lower $n - k$ singular values [AL86]. However, the cost of such methods would still be $\tilde{O}(n^2 k)$, which our work improves on. There has also been work on iterative methods for the conceptually related problem of solving poorly conditioned linear system consisting of multiple well-conditioned subspaces [KMS+22], although the challenges are different.

Also related to our work is recent progress on analyzing and improving *stochastic iterative methods* like stochastic gradient descent, randomized Kaczmarz, stochastic coordinate descent, and variants thereof [SV09, RSB12, JZ13, SSZ14]. A major development in the area is that it is possible to obtain runtimes for solving linear systems that depend on the *average condition* number, $\bar{\kappa} = \frac{1}{n}\sum_{i=1}^{n} \frac{\sigma_i}{\sigma_n}$ instead of the standard condition number $\kappa = \frac{\sigma_1}{\sigma_n}$. E.g., for PD linear systems, [LS13] presents a variant of coordinate descent that runs in $\tilde{O}(n^2\sqrt{\bar{\kappa}})$ time vs. $\tilde{O}(n^2\sqrt{\kappa})$ for the conjugate gradient method. Similar results have been proven for variants of stochastic gradient descent [FGKS15]. More recently, [DLNR24] uses a connection between block coordinate descent and low-rank approximation [DR24], obtaining an algorithm that runs in $\tilde{O}(n^2\sqrt{\bar{\kappa}_\ell})$ time for $\ell = O(n^{\frac{1}{\omega-1}})$, where $\bar{\kappa}_\ell = \frac{1}{n-\ell}\sum_{i>\ell} \frac{\sigma_i}{\sigma_n} \leq \bar{\kappa}$. Naturally, these average condition numbers $\bar{\kappa}$ and $\bar{\kappa}_\ell$ can be significantly smaller than κ if $\mathbf{A}$ only has a few large singular values and is otherwise well-conditioned. Perhaps unsurprisingly then, the current state-of-the-art methods for Problem 1.1 and downstream applications like matrix norm approximation rely on stochastic iterative methods [MNS+18, DY24, DLNR24]. A high-level observation of our work is that it is possible to match and actually exceed the efficiency of these methods using deterministic iterative methods with randomized preconditioning. This will become more apparent in Section 3, where our main theorem is stated in terms of a dependence on an average condition number.

2 Preliminaries

Notation. Throughout this paper, we use $\mathcal{S}_n^+$ to denote the positive semidefinite (PSD) cone and use $\mathcal{S}_n^{++}$ to denote the positive definite (PD) cone. For vector $\mathbf{x}$, we use $\|\mathbf{x}\|$ to denote its Euclidean norm, and for a PSD matrix $\mathbf{A}$, we denote $\|\mathbf{x}\|_\mathbf{A} := \sqrt{\mathbf{x}^\top \mathbf{A}\mathbf{x}}$. For matrix $\mathbf{A}$ with singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$, we let $\|\mathbf{A}\| = \sigma_1$ and $\|\mathbf{A}\|_F$ denote the operator norm and Frobenius norm, respectively. We let $\kappa(\mathbf{A}) = \sigma_1/\sigma_n$ denote its condition number. We let $\mathbf{A}^\dagger = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$ denote the pseudoinverse of $\mathbf{A}$. We use $\mathrm{nnz}(\mathbf{A})$ to denote the number of non-zero entries of $\mathbf{A}$. For two $n \times n$ PSD matrices $\mathbf{A}, \mathbf{B}$ and any $\epsilon > 0$, we say $\mathbf{A} \approx_{1+\epsilon} \mathbf{B}$ if $\frac{1}{1+\epsilon}\mathbf{A} \preceq \mathbf{B} \preceq (1+\epsilon)\mathbf{A}$ holds, where $\preceq$ denotes the matrix Loewner order. In the following sections, for matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and failure probability $\delta > 0$, we use $\tilde{O}(\cdot)$ to omit $\mathrm{polylog}(mn/\delta)$ factors[4]. In this paper, by saying "with high probablity" we mean with probability at least $1 - 1/\mathrm{poly}(n)$.

For matrix $\mathbf{A} \in \mathcal{S}_n^+$, let $\{\lambda_i\}_{i=1}^n$ be its eigenvalues in non-increasing order. For any $\lambda > 0$, we define the effective dimension of $\mathbf{A}$ as $d_\lambda := \mathrm{tr}(\mathbf{A}(\mathbf{A} + \lambda\mathbf{I})^{-1}) = \sum_{i=1}^n \frac{\lambda_i}{\lambda_i + \lambda}$. We will require the following basic lemma on the effective dimension:

LEMMA 2.1. (LEMMA 5.4 IN [FTU23]) *For $\mathbf{A} \in \mathcal{S}_n^+$ and $\lambda > 0$, we have the following holds:*

1. *For any $\gamma > 0$, if $j \geq (1 + \gamma^{-1})d_\lambda$, then $\lambda_j \leq \gamma\lambda$.*
2. *If $k \geq d_\lambda$, then $\sum_{i>k} \lambda_i \leq \lambda \cdot d_\lambda$.*

Sparse Subspace Embedding. Following [CNW16, Coh16], we define the sparse embedding matrix $\mathbf{S} \in \mathbb{R}^{s\times n}$ with sparsity parameter γ, which will be used in our construction of the Nyström preconditioner, as follows. Notice that given $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{SA}$ can be computed in time $O(\gamma \cdot \mathrm{nnz}(\mathbf{A}))$.

[4]Notice that in comparison, in Section 1 we use $\tilde{O}(\cdot)$ to hide all log factors for conciseness.

DEFINITION 2.1. (SPARSE EMBEDDING MATRIX) *We define an $s \times n$ matrix $\mathbf{S}$ to be a sparse embedding matrix with sparsity γ, if the columns of $\mathbf{S}$ are independent, and for each column of $\mathbf{S}$, there are γ random entries chosen uniformly without replacement and set to $\pm 1/\sqrt{\gamma}$ independently, with other entries in that column being set to 0.*

Based on the construction of sparse embedding matrix, recent work [CDDR24] further constructs a fast oblivious subspace embedding (OSE) matrix and shows an optimal $O(d)$ result for embedding dimension, as stated in the following lemma. Throughout this paper, we will use $\mathbf{\Phi}$ to denote this oblivious embedding matrix, which will be used in the construction of a preconditioner for the second level of linear system solving.

LEMMA 2.2. (ADAPTED FROM THEOREM 1.4 IN [CDDR24]) *Given $\mathbf{A} \in \mathbb{R}^{n\times d}, \epsilon < 1/2$, $\delta < 1/2$, in time $O(\mathrm{nnz}(\mathbf{A})\log(d/\delta)/\epsilon + d^2\log^4(d/\delta)/\epsilon^6)$ we can compute $\mathbf{\Phi A}$ where $\mathbf{\Phi} \in \mathbb{R}^{\phi\times n}$ is an embedding matrix with $\phi = O((d+\log(1/\delta))/\epsilon^2)$, such that with probability $1-\delta$ we have*

$$\forall \mathbf{x} \in \mathbb{R}^d, \quad \frac{1}{1+\epsilon}\|\mathbf{Ax}\| \le \|\mathbf{\Phi Ax}\| \le (1+\epsilon)\|\mathbf{Ax}\|.$$

Computational Model. Our results are proven in the real RAM model (i.e, exact arithmetic). We assume the inputs $\mathbf{A}$ and $\mathbf{b}$ are given with real-valued entries, and basic arithmetic operations $(+, -, \times, \div, \sqrt{\cdot})$ can be performed exactly on real numbers in $O(1)$ time. We still need to leverage results on the finite-precision behavior of iterative solvers like the Lanczos method [Pai76, MMS18] to account for the fact that we apply preconditioners inexactly, even in the real RAM model. A full analysis of our methods in a finite precision model of computation is beyond the scope of this work, but is an important next step for future work on Multi-level Sketched Preconditioning.

3 Main Technical Results

In this section, we present our main technical result in its full generality, and then discuss how it can be used to recover the claims in Section 1. In this result, we address the task of solving a linear system of the form $(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})\mathbf{x} = \mathbf{c}$ for an $m \times n$ matrix $\mathbf{A}$ and $\lambda \ge 0$. Note that the regularization parameter λ is entirely optional (since we can set $\lambda = 0$), and it is included here primarily for the sake of applications to kernel ridge regression and Schatten norm estimation. To recover the setting from Theorem 1.1, i.e., solving $\mathbf{Ax} = \mathbf{b}$, we can simply rewrite this problem via the normal equations as $\mathbf{A}^\top\mathbf{Ax} = \mathbf{A}^\top\mathbf{b}$, then set $\mathbf{c} = \mathbf{A}^\top\mathbf{b}$ and $\lambda = 0$.

THEOREM 3.1. (MAIN TECHNICAL RESULT) *Given $\mathbf{A} \in \mathbb{R}^{m\times n}$ with condition number κ, vector $\mathbf{b} \in \mathbb{R}^n$ and regularization parameter $\lambda \ge 0$, let $\{\sigma_i\}_{i=1}^n$ be the singular values of $\mathbf{A}$ in decreasing order, and let $\mathbf{x}^* = (\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{c}$. For any $l \in \{\log n + 1, \ldots, n\}$, define $\bar{\kappa}_{l,\lambda} := (\frac{1}{n-l}\sum_{i>l}\sigma_i^2/(\sigma_n^2+\lambda))^{1/2}$. Given $\epsilon > 0$ and $\delta \in (0, 1/8)$, if Algorithm 3 is run with $\lambda_0 = \frac{2}{l}\sum_{i>l}\sigma_i^2$, $s = O(l\log(l/\delta))$, $\gamma = O(\log(l/\delta))$, $\phi = O(s+\log 1/\delta)$ and $t_{\max} = O(\sqrt{n/l}\cdot\bar{\kappa}_{l,\lambda}\log(\bar{\kappa}_{l,\lambda}/\epsilon))$, then with probability at least $1-\delta$, it will output $\tilde{\mathbf{x}}$ such that $\|\tilde{\mathbf{x}}-\mathbf{x}^*\|_{\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I}} \le \epsilon\|\mathbf{x}^*\|_{\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I}}$ in time*

$$\tilde{O}\left(\mathrm{nnz}(\mathbf{A})\sqrt{\frac{n}{l}}\bar{\kappa}_{l,\lambda}\log^3(\kappa/\epsilon) + l^\omega\right)$$

where $\tilde{O}(\cdot)$ hides $\mathrm{polylog}(mn/\delta)$ factors. Moreover, if instead of matrix $\mathbf{A}$, we are given the matrix $\mathbf{A}^\top\mathbf{A}$ directly, then the same time complexity can be achieved with $\mathrm{nnz}(\mathbf{A})$ replaced by $\mathrm{nnz}(\mathbf{A}^\top\mathbf{A})$ by using Algorithm 1. See Theorem 4.1 for details.

REMARK 3.1. *The quantity $\bar{\kappa}_{l,\lambda}$ above has appeared (in similar forms) in prior work on the convergence of stochastic iterative methods for linear systems [LS13, MNS+18, DY24]. When $\lambda = 0$, this quantity satisfies $1 \le \bar{\kappa}_{l,0} \le \sigma_{l+1}/\sigma_n$, and it behaves like a typical averaged condition number of $\mathbf{A}$. However, when $\lambda > 0$, $\bar{\kappa}_{l,\lambda}$ can in fact be much less than 1, which we exploit in the application to kernel ridge regression.*

We next demonstrate how Theorem 3.1 can be used to show our main result for matrices with k large singular values. To do so, we will optimize over the choice of l as follows.

Proof. [Proof of Theorem 1.1] As mentioned above, to solve the consistent linear system $\mathbf{Ax} = \mathbf{b}$, we can apply Theorem 3.1 with choice $\lambda = 0$ and $\mathbf{c} = \mathbf{A}^\top\mathbf{b}$. The solution $\mathbf{x}^*$ of the resulting linear system $\mathbf{A}^\top\mathbf{Ax} = \mathbf{A}^\top\mathbf{b}$,

$(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b} = \mathbf{A}^\dagger\mathbf{b}$, is the minimum norm solution of the general linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$. Additionally, the output guarantee from Theorem 3.1, that $\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_{\mathbf{A}^\top\mathbf{A}} \leq \epsilon\|\mathbf{x}^*\|_{\mathbf{A}^\top\mathbf{A}}$, ensures that $\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\| \leq \epsilon\|\mathbf{b}\|$ as required by Theorem 1.1.

It remains to obtain the desired time complexity by optimizing over l, which we will choose to be larger than k. Notice that for $l > k$, we have $\bar{\kappa}_{l,0} \leq \sigma_{l+1}/\sigma_n = O(1)$ under the assumption of Theorem 1.1. So, upper bounding $\text{nnz}(\mathbf{A}) \leq n^2$, we can upper bound the runtime in Theorem 3.1 by:

$$\tilde{O}(n^{2.5}/\sqrt{l} + l^\omega), \quad \text{for any } l \geq k.$$

Optimizing over l so that both terms are proportionally large, we obtain the following two cases:

- If $k \leq n^{5/(2\omega+1)}$, then we choose $l = n^{5/(2\omega+1)}$ and obtain time complexity $\tilde{O}(l^\omega) = \tilde{O}(n^{2+\frac{\omega-2}{2\omega+1}})$.
- If $n^{5/(2\omega+1)} < k < n$, then we choose $l = k$, and obtain time complexity $\tilde{O}(l^\omega) = \tilde{O}(k^\omega)$.

Thus, the overall time complexity becomes:

$$\tilde{O}(n^{2+\frac{\omega-2}{2\omega+1}} + k^\omega),$$

which, using current matrix multiplication exponent $\omega \approx 2.372$, simplifies to $\tilde{O}(n^{2.065} + k^\omega)$. ∎

3.1 Applications to Regularized Linear Systems and Least Squares Some of the most important applications of linear systems, particularly in the context of machine learning and statistics, are regularized and unregularized least squares problems. In this section, we show how Theorem 3.1 can be adapted and applied to these settings.

Kernel ridge regression. One of the most computationally expensive variants of these tasks arises when we consider kernel-based learning methods [RCR17, MB17, MCRR20]. These approaches work by implicitly constructing expanded high-dimensional representations of data points, which are accessed only through inner products, which are computed using a kernel function. For a dataset with n datapoints, this gives rise to an $n \times n$ positive definite kernel matrix $\mathbf{K}$, whose (i, j) entry is the inner product between the ith and jth data point in the expanded feature representation. The resulting prediction model forms the kernel ridge regression (KRR) task:

$$\min_{\mathbf{x}} \frac{1}{n}\sum_{i=1}^{n}([\mathbf{K}\mathbf{x}]_i - y_i)^2 + \frac{\lambda}{2}\mathbf{x}^\top\mathbf{K}\mathbf{x},$$

where $[\mathbf{K}\mathbf{x}]_i$ is the ith entry of the length n vector $\mathbf{K}\mathbf{x}$ and $y_1, \ldots, y_n$ are training labels. KRR is equivalent to solving the regularized positive definite linear system $(\mathbf{K} + n\lambda\mathbf{I})\mathbf{x} = \mathbf{y}$. Since $\mathbf{K}$ is $n \times n$, the cost of solving this problem exactly scales with $O(n^\omega)$. Regularized linear systems of this form also arise independently in continuous second-order optimization methods, where instead of the kernel matrix $\mathbf{K}$, we consider a Hessian matrix $\mathbf{H}$, and the regularization term λ is a parameter of the optimization algorithm [Lev44, Mar63, MS12, JJM23].

We show how to adapt our main result to the setting above, giving a faster method for solving any linear system of the form $(\mathbf{A} + \lambda\mathbf{I})\mathbf{x} = \mathbf{b}$ for a positive semidefinite matrix $\mathbf{A}$. Our method improves on the previously best known time complexity of this problem [ACW17, FTU23], from $\tilde{O}(n^2 + nd_\lambda{}^{\omega-1})$ to $\tilde{O}(n^2 + d_\lambda{}^\omega)$, where recall that d_λ is the λ-effective dimension of $\mathbf{A}$. We note that this result takes advantage of the fact that $\bar{\kappa}_{l,\lambda}$ can in fact be smaller than 1 for $\lambda > 0$.

THEOREM 3.2. (REGULARIZED LINEAR SYSTEMS, FORMAL VERSION OF THEOREM 1.2) *Consider positive semidefinite $\mathbf{A} \in \mathcal{S}_n^+$ with condition number κ and effective $d_\lambda = \text{tr}(\mathbf{A}(\mathbf{A} + \lambda\mathbf{I})^{-1})$ for $\lambda > 0$. For a target $\mathbf{b} \in \mathbb{R}^n$, let $\mathbf{x}^* = (\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{b}$. Given $\epsilon > 0$ and $0 < \delta < 1/8$, with probability $1 - \delta$ we can compute $\tilde{\mathbf{x}}$ such that $\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_{\mathbf{A}+\lambda\mathbf{I}} \leq \epsilon\|\mathbf{x}^*\|_{\mathbf{A}+\lambda\mathbf{I}}$ in time*

$$\tilde{O}\left(n^2 \cdot \log^3(\kappa/\epsilon) + d_\lambda{}^\omega\right).$$

Proof. Without loss of generality, we assume that $d_\lambda < n/4$. If d_λ is larger, the stated runtime follows by simply using a direct $O(n^\omega)$ time method for inverting $(\mathbf{A} + \lambda\mathbf{I})$.

We obtain the result by applying Theorem 3.1 to the matrix $\mathbf{A}^{1/2}$. As stated in the last lines of the theorem, it holds even if we only have access to $(\mathbf{A}^{1/2})^T\mathbf{A}^{1/2} = \mathbf{A}$. We show that if we choose $l = 2d_\lambda$, we have $\bar{\kappa}_{l,\lambda}^2(\mathbf{A}^{1/2}) = O(l/n)$. In particular, recall that $d_\lambda = \mathrm{tr}(\mathbf{A}(\mathbf{A}+\lambda\mathbf{I})^{-1})$ and, from Lemma 2.1 we know that if we choose $l = 2d_\lambda$ then $\lambda_l(\mathbf{A}) \leq \lambda$. We thus have:

$$l = 2d_\lambda = 2\sum_{i=1}^{n}\frac{\lambda_i(\mathbf{A})}{\lambda_i(\mathbf{A})+\lambda} > 2\sum_{i=l}^{n}\frac{\lambda_i(\mathbf{A})}{\lambda_i(\mathbf{A})+\lambda} \geq \frac{2\cdot\sum_{i>l}\lambda_i(\mathbf{A})}{\lambda_l(\mathbf{A})+\lambda} \geq \frac{1}{\lambda}\sum_{i>l}\lambda_i(\mathbf{A}).$$

By applying this result we have

$$\bar{\kappa}_{l,\lambda}^2(\mathbf{A}^{1/2}) = \frac{1}{n-l}\sum_{i>l}\frac{\lambda_i(\mathbf{A})}{\lambda_n(\mathbf{A})+\lambda} \leq \frac{1}{n-l}\cdot\frac{l\lambda}{\lambda_n(\mathbf{A})+\lambda} \leq \frac{l}{n-l} \leq \frac{2l}{n}.$$

In the last step we use that, since $d_\lambda \leq n/4$ by assumption, $l \leq n/2$.

Now we simply plug into Theorem 3.1, noting that $\sqrt{\frac{n}{l}}\bar{\kappa}_{l,\lambda}$ can be upper bounded by $\sqrt{2}$ given the result above. We conclude that we can compute $\tilde{\mathbf{x}}$ such that $\|\tilde{\mathbf{x}}-\mathbf{x}^*\|_{\mathbf{A}+\lambda\mathbf{I}} \leq \epsilon\|\mathbf{x}^*\|_{\mathbf{A}+\lambda\mathbf{I}}$ in time $\tilde{O}(n^2\cdot\log^3(\kappa/\epsilon)+{d_\lambda}^\omega)$. ∎

Least Squares. Our Theorem 3.1 naturally extends to over-determined linear systems. For simplicity we state a result in the setting of Problem 1.1, where $\mathbf{A}$ has at most k large singular values and is otherwise well conditioned.

COROLLARY 3.1. (LEAST SQUARES) *Given matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ with at most k singular values larger than $O(1)$ times its smallest singular value, $\mathbf{b} \in \mathbb{R}^n, \epsilon > 0$ and $0 < \delta < 1/8$, with probability at least $1-\delta$, we can compute $\tilde{\mathbf{x}}$ 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 time*

$$\tilde{O}\left((\mathrm{nnz}(\mathbf{A}) + d^{2.065})\cdot\log 1/\epsilon + k^\omega\right).$$

This result is obtained by combining our MSP method with a standard sketch-and-precondition iterative solver [RT08,Epp24]. In particular, for a tall matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ where $n \gg d$, one can first construct a sketch $\tilde{\mathbf{A}}$ with $O(d)$ rows using a constant-factor sparse oblivious subspace embedding (e.g., Lemma 2.2) which takes $\tilde{O}(\mathrm{nnz}(\mathbf{A})+d^2)$ time. To solve a least squares problem involving $\mathbf{A}$, it suffices to solve $O(\log(1/\epsilon))$ linear systems of the form $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}\mathbf{g}$ for some vector $\mathbf{g}$ up to constant error. See e.g. Section 10 of [DY24] for details.

To solve these required linear systems, we apply Theorem 3.1 directly. We note that, since $\tilde{\mathbf{A}}$ is a subspace embedding of $\mathbf{A}$, all of its singular values are within a multiplicative constant factor of those of $\mathbf{A}$. Accordingly, it too has at most k singular values greater than a constant times its smallest singular value. We conclude that, for $l \geq k$, each application of Theorem 3.1 takes $\tilde{O}(\mathrm{nnz}(\mathbf{A})+d^2\sqrt{d/l})$ time. The additive $O(l^\omega)$ term in the runtime of Theorem 3.1 comes from the construction of an inner preconditioner for $\tilde{\mathbf{A}}$, so is only incurred a single time. Optimizing over the choice of l yields Corollarly 3.1. Notice that we only incur a $\log 1/\epsilon$ dependence in the result instead of $\log^3 1/\epsilon$. This is because each system involving $\tilde{\mathbf{A}}$ only needs to be solved to constant accuracy for the overall sketch-and-precondition method to converge in $O(\log 1/\epsilon)$ iterations.

3.2 Applications to Matrix Norm Estimation We next discuss how our methods can be used in the task of estimating the Schatten norm of a matrix, providing a direct improvement over a result obtained in [MNS+18]. For the sake of simplicity, we focus on the case of a dense square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$. However, using similar strategies to those described in [MNS+18], our results can also be used to improve time complexities when $\mathbf{A}$ is sparse or rectangular.

To obtain our results, we note that the norm estimation algorithms of [MNS+18] are in fact meta-algorithms that require a black-box linear solver for regularized systems. Therefore any improvement to the time complexity of the black-box solver leads to a potential improvement in the complexity of Schatten norm estimation (as well as to other spectrum approximation tasks discussed in that paper). The key black-box requirement from [MNS+18] is as follows: Given a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, parameter $\lambda > 0$, tolerance $\epsilon \in (0,1)$, and a vector $\mathbf{y} \in \mathbb{R}^n$, return a vector $\mathbf{x}$ such that:

$$\|\mathbf{x}-\mathbf{M}_\lambda^{-1}\mathbf{y}\|_{\mathbf{M}_\lambda} \leq \epsilon\|\mathbf{y}\|_{\mathbf{M}_\lambda^{-1}}, \quad \text{where} \quad \mathbf{M}_\lambda = \mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I}. \tag{3.1}$$

A key parameter in the analysis of [MNS+18] for solving such linear systems turns out to be the same averaged tail condition number that arises in our analysis (up to adjustments in notation):

$$\bar{\kappa}_{k,\lambda}(\mathbf{A}) \coloneqq \Big(\frac{1}{n-k}\sum_{i>k}\frac{\sigma_i^2(\mathbf{A})}{\sigma_{\min}^2(\mathbf{A})+\lambda}\Big)^{1/2}.$$

In particular, the main result of [MNS+18] for Schatten norm estimation can be reformulated as follows:

LEMMA 3.1. (ADAPTED FROM THE PROOF OF COROLLARY 12 IN [MNS+18]) *For any matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, parameter $p \in (0,2)$, $\epsilon \in (0,1)$, and a parameter $k \le n$, there is an algorithm that, using $\tilde{O}(\frac{1}{\epsilon^5 p})$ calls to an $n \times n$ ridge regression problem (3.1) with $\bar{\kappa}_{k,\lambda} \le \frac{1}{\epsilon^{1/p}}(\frac{n}{k})^{1/p-1/2}$, returns $X \in (1\pm\epsilon)\|\mathbf{A}\|_p^p$.*

We note that a similar statement applies to the general spectral sums approximation result of [MNS+18] (Theorem 11), which can also by used to approximate other norms like the Ky Fan and Orlicz norm. We focus on the Schatten norms merely for the sake of simplicity and conciseness. Also, we only focus on Schatten norms with $0 < p < 2$, including the most important case of $p = 1$, as the existing algorithms for the case of $p \ge 2$ are already near-optimal.

We are now ready to state our result for Schatten norm estimation, by combining Lemma 3.1 with our linear system solver.

THEOREM 3.3. *For any $p \in (0,2)$ and $\mathbf{A} \in \mathbb{R}^{n\times n}$ there is an algorithm that, with high probability, returns $X \in (1\pm\epsilon)\|\mathbf{A}\|_p^p$ which runs in time:*

$$\tilde{O}\Big(n^{2+\frac{\omega-2}{p\omega+1}}\mathrm{poly}(1/\epsilon)\Big).$$

In particular, for $p = 1$, we can get a constant factor approximation to the Schatten 1-norm of $\mathbf{A}$ in $\tilde{O}(n^{2.11})$ time with fast matrix multiplication and in $\tilde{O}(n^{2.25})$ time without fast matrix multiplication.

REMARK 3.2. *This result improves the exponent of n in the time complexity, compared to [MNS+18], for any $0 < p < 1.5$. In particular, for $p = 1$, we improve from $\tilde{O}(n^{2.18})$ to $\tilde{O}(n^{2.11})$ with fast matrix multiplication, and from $\tilde{O}(n^{2.33})$ to $\tilde{O}(n^{2.25})$ without fast matrix multiplication. While we did not optimize the time complexity dependence on ϵ, it should be possible to recover the $O(\frac{1}{\epsilon^{\max\{3,1+1/p\}}})$ dependence from [MNS+18] by carefully repeating their optimized analysis given in Theorem 31.*

Proof. We will prove this result by applying our main result, Theorem 3.1. However, observe that this theorem has a logarithmic dependence on the condition number of $\mathbf{A}$, which we would like to avoid. To do so, observe that we can reduce to the case where the input $\mathbf{A}$ has condition number $\mathrm{poly}(n/\epsilon)$. If it does not, we can simply add $z\mathbf{I}$ to $\mathbf{A}$, where $\mathbf{I}$ is an $n \times n$ identity matrix and z is a mean zero Gaussian random variable with standard deviation $\|\mathbf{A}\| \cdot \mathrm{poly}(\epsilon/n)$. $\|\mathbf{A}\|$ can be approximated to multiplicative accuracy in $\tilde{O}(n^2)$ time using standard power method. With high probability, we will have that $|z| \le \|\mathbf{A}\| \cdot \mathrm{poly}(\epsilon/n)$, so the Schatten p-norm of $\mathbf{A}+z\mathbf{I}$ is within a multiplicative $(1\pm\epsilon/2)$ factor of that of $\mathbf{A}$. Moreover, by a union bound and standard anti-concentration of Gaussian random variables, we will have that all eigenvalues of $\mathbf{A} + z\mathbf{I}$ are at least $\|\mathbf{A}\| \cdot \mathrm{poly}(\epsilon/n)$ far away from zero with high probability, leading to condition number $\mathrm{poly}(n/\epsilon)$.

With the condition number bounded, for any $\lambda > 0$ and k, using our main linear system solver from Theorem 3.1, we can solve the regularized linear system (3.1) in time $\tilde{O}(\mathrm{nnz}(\mathbf{A})\sqrt{\frac{n}{k}}\bar{\kappa}_{k,\lambda} + k^\omega)$. Focusing on dense matrices so that $\mathrm{nnz}(\mathbf{A}) = n^2$, and assuming that $\bar{\kappa}_{k,\lambda} \le \frac{1}{\epsilon^{1/p}}(\frac{n}{k})^{1/p-1/2}$ as in Lemma 3.1, we obtain the following overall time complexity for Schatten norm estimation:

$$\tilde{O}\left(\frac{1}{p\epsilon^5}n^2\sqrt{\frac{n}{k}}\epsilon^{-1/p}\Big(\frac{n}{k}\Big)^{1/p-1/2} + k^\omega\right) = \tilde{O}\Big(\frac{1}{p\epsilon^{5+1/p}}\frac{n^{2+1/p}}{k^{1/p}} + k^\omega\Big),$$

where we use the fact that the $\tilde{O}(k^\omega)$ (needed to build our recursive preconditioner) only has to be performed once and can be reused for all linear solves. Now, it remains to optimize the time complexity over k. Without optimizing over ϵ dependence, we simply balance out the two terms $\frac{n^{2+1/p}}{k^{1/p}}$ and k^ω, obtaining $n^{\frac{2p\omega+\omega}{p\omega+1}} = n^{2+\frac{\omega-2}{p\omega+1}}$. ∎

4 Two-Level MSP for Positive Definite Linear Systems

Before proceeding into the proof of our main result Theorem 3.1, we first consider a special (and easier) case of solving $(\mathbf{A}+\lambda\mathbf{I})\mathbf{x}=\mathbf{b}$, where $\mathbf{A}\in\mathcal{S}_n^{++}$ and $\lambda\geq 0$. Let $\{\lambda_i\}_{i=1}^n$ be the eigenvalues of $\mathbf{A}$ in decreasing order. Given $l<n$ and $\lambda\geq 0$, we define the average regularized tail condition number of $\mathbf{A}$ as $\bar{\kappa}_{l,\lambda}(\mathbf{A}) := \frac{1}{n-l}\sum_{i>l}^{n}\frac{\lambda_i}{\lambda_n+\lambda}$. Notice that $\bar{\kappa}_{l,\lambda}(\mathbf{A})$ is decreasing in both l and λ. For simplicity, we denote it as $\bar{\kappa}_l$ if we assume that $\lambda=0$ and matrix $\mathbf{A}$ is clear from the context.

We solve this positive definite linear system with preconditioned Lanczos method. The following Lemma 4.1 guarantees that, when using a preconditioner $\mathbf{M}$, if for given vector $\mathbf{r}$ we can approximate $\mathbf{M}^{-1}\mathbf{r}$ well enough and satisfy Eq.(4.1), then preconditioned Lanczos can converge in roughly the same number of iterations as if we compute $\mathbf{M}^{-1}\mathbf{r}$ exactly. Details are in Section 6.

LEMMA 4.1. (PRECONDITIONED LANCZOS, RESTATED THEOREM 6.1) *Consider solving* $\mathbf{Ax}=\mathbf{b}$ *for positive definite* $\mathbf{A}$ *using preconditioned Lanczos provided with a function* SolveM *that, for some preconditioner* $\mathbf{M}$ *and any vector* $\mathbf{r}$ *returns*

$$\|\mathsf{SolveM}(\mathbf{r})-\mathbf{M}^{-1}\mathbf{r}\|_{\mathbf{M}}\leq\epsilon_0\cdot\|\mathbf{M}^{-1}\mathbf{r}\|_{\mathbf{M}}. \tag{4.1}$$

If $\epsilon_0\leq(\frac{\epsilon}{\kappa_{\mathbf{M}}n})^c$ *for a fixed constant* $c>0$*, where* $\kappa_{\mathbf{M}}$ *is the condition number of* $\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$*, Then, preconditioned Lanczos (Algorithm 6) with* $t=O(\sqrt{\kappa_{\mathbf{M}}}\log(\kappa_{\mathbf{M}}/\epsilon))$ *iterations returns* $\tilde{\mathbf{x}}$ *s.t.:*

$$\|\tilde{\mathbf{x}}-\mathbf{x}^*\|_{\mathbf{A}}\leq\epsilon\cdot\|\mathbf{x}^*\|_{\mathbf{A}}\quad \textit{where}\quad \mathbf{x}^*=\mathbf{A}^{-1}\mathbf{b}.$$

The construction of preconditioner $\mathbf{M}$ is based on Nyström approximation. Let $\mathbf{C}=\mathbf{A}\mathbf{S}^\top$ and $\mathbf{W}=\mathbf{S}\mathbf{A}\mathbf{S}^\top$ where $\mathbf{S}\in\mathbb{R}^{s\times n}$ is some sketching matrix. Then the Nyström approximation of $\mathbf{A}$ can be expressed as $\hat{\mathbf{A}}_{\text{nys}}=\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top$. We show that if we choose $\mathbf{S}$ to be a sparse embedding matrix (according to Definition 2.1) with $s=\tilde{O}(l)$ rows and $\gamma=\tilde{O}(1)$ non-zero entries per column, then we have the approximation guarantee $\|\hat{\mathbf{A}}_{\text{nys}}-\mathbf{A}\|\leq\frac{2}{l}\sum_{i>l}\lambda_i$, which in turn gives $\kappa_{\mathbf{M}}=O(\bar{\kappa}_{l,\lambda}n/l)$ for $\mathbf{M}:=\hat{\mathbf{A}}_{\text{nys}}+\tilde{\lambda}\mathbf{I}$ where $\tilde{\lambda}:=\lambda+\frac{2}{l}\sum_{i>l}\lambda_i$, see Lemma 4.2. With the choice of $\mathbf{M}$ we can verify the following holds:

$$\mathbf{M}^{-1}\mathbf{r}=\frac{1}{\tilde{\lambda}}\left(\mathbf{r}-\mathbf{C}(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\mathbf{r}\right).$$

To approximate $\mathbf{M}^{-1}\mathbf{r}$ we can first solve the linear system $(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})\mathbf{y}=\mathbf{C}^\top\mathbf{r}$ and get approximate solution $\hat{\mathbf{y}}$, then use $\hat{\mathbf{w}}:=\frac{1}{\tilde{\lambda}}(\mathbf{r}-\mathbf{C}\hat{\mathbf{y}})$ as the approximator, see Lemma 4.3. However, constructing such a linear system takes $O(ns^2)$ which is unaffordable for us. Instead, we introduce a second level, and again use preconditioned Lanczos as the Level-2 solver to solve the linear system $(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})\mathbf{y}=\mathbf{C}^\top\mathbf{r}$ without explicitly computing $\mathbf{C}^\top\mathbf{C}$. Compared with the first level where we do the preconditioning coarsely, in the second level we obtain an optimal $O(1)$ condition number after preconditioning.

Let $\mathbf{\Phi}\in\mathbb{R}^{\phi\times n}$ be the oblivious subspace embedding matrix from Lemma 2.2 with $\phi=\tilde{O}(s)$. We can construct the Level-2 preconditioner as $\mathbf{M}_2:=\tilde{\mathbf{C}}^\top\tilde{\mathbf{C}}+\lambda\mathbf{W}$ where $\tilde{\mathbf{C}}=\mathbf{\Phi}\mathbf{C}$ in time $\tilde{O}(ns+s^\omega)$. Notice that this step only needs to be done once. By transforming the error bound (that we get from Level-2 preconditioned Lanczos) from $\hat{\mathbf{y}}$ to $\hat{\mathbf{w}}$ using Lemma 4.3, we can verify the assumption Eq.(4.1) which is required in Level-1 preconditioned Lanczos, thus finishing the proof. Formally, we have the following theorem for solving positive definite linear systems, where note that the condition number $\bar{\kappa}_{l,\lambda}$ is defined differently in terms of the eigenvalues of $\mathbf{A}$, rather than in terms of the squared singular values. Also note that although we assume $\mathbf{A}\in\mathcal{S}_n^{++}$, indeed we only need $\mathbf{A}\in\mathcal{S}_n^{+}$ and $\mathbf{A}+\lambda\mathbf{I}\in\mathcal{S}_n^{++}$ (which is satisfied if $\lambda>0$).

THEOREM 4.1. (MSP, POSITIVE DEFINITE) *Given* $\mathbf{A}\in\mathcal{S}_n^{++}$ *with condition number* κ*,* $\mathbf{b}\in\mathbb{R}^n$ *and regularized term* $\lambda\geq 0$*. Let* $\{\lambda_i\}_{i=1}^n$ *be the eigenvalues of* $\mathbf{A}$ *in decreasing order, let* $\mathbf{x}^*=(\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{b}$ *be the solution of the regularized linear system* $(\mathbf{A}+\lambda\mathbf{I})\mathbf{x}=\mathbf{b}$*. For any* $\log n<l<n$*, we define* $\bar{\kappa}_{l,\lambda}:=\frac{1}{n-l}\sum_{i>l}^{n}\lambda_i/(\lambda_n+\lambda)$*. Given* $\epsilon>0$ *and* $\delta<1/8$*, with probability at least* $1-\delta$*, running Algorithm 1 with choice* $\lambda_0=\frac{2}{l}\cdot\sum_{i>l}\lambda_i$, $s=O(l\log(l/\delta))$, $\gamma=O(\log(l/\delta))$, $\phi=O(s+\log 1/\delta)$ *and* $t_{\max}=O(\sqrt{\bar{\kappa}_{l,\lambda}n/l}\cdot\log(\bar{\kappa}_{l,\lambda}n/\epsilon))$ *will output* $\tilde{\mathbf{x}}$ *such that* $\|\tilde{\mathbf{x}}-\mathbf{x}^*\|_{\mathbf{A}+\lambda\mathbf{I}}\leq\epsilon\|\mathbf{x}^*\|_{\mathbf{A}+\lambda\mathbf{I}}$ *in time*

$$\tilde{O}\left(\text{nnz}(\mathbf{A})\sqrt{\frac{\bar{\kappa}_{l,\lambda}n}{l}}\log^2(\kappa/\epsilon)+l^\omega\right)$$

where $\tilde{O}(\cdot)$ hides polylog(n/δ) *factors.*

Algorithm 1 MSP for solving regularized positive definite linear system $(\mathbf{A}+\lambda\mathbf{I})\mathbf{x}=\mathbf{b}$.

Input: matrix $\mathbf{A}\in\mathcal{S}_n^{++}$, regularizer terms λ and λ_0, vector $\mathbf{b}\in\mathbb{R}^n$, sparse sketch size s, # of non-zeros γ, level-2 sketch size ϕ, # of iterations $t_{\max}$;
Construct sparse embedding matrix $\mathbf{S}\in\mathbb{R}^{s\times n}$ with γ non-zeros per column; ▷ Def. 2.1.
Compute $\tilde{\lambda}=\lambda+\lambda_0$, $\mathbf{C}=\mathbf{A}\mathbf{S}^\top$, $\mathbf{W}=\mathbf{S}\mathbf{C}$;
Compute $\tilde{\mathbf{C}}=\mathbf{\Phi}\mathbf{C}\in\mathbb{R}^{\phi\times s}$ and $\mathbf{M}_2^{-1}=(\tilde{\mathbf{C}}^\top\tilde{\mathbf{C}}+\tilde{\lambda}\mathbf{W})^{-1}$; ▷ Lemma 2.2.
Solve $\tilde{\mathbf{x}}$ by calling preconditioned Lanczos with $(\mathbf{A}+\lambda\mathbf{I},\mathbf{b},\mathsf{SolveM1},t_{\max})$; ▷ Alg. 6.
return $\tilde{\mathbf{x}}$; ▷ Solves $(\mathbf{A}+\lambda\mathbf{I})\mathbf{x}=\mathbf{b}$.

Algorithm 2 Level-1 auxiliary function SolveM1 for solving $\mathbf{M}\mathbf{w}=\mathbf{r}$ (positive definite case).

function SolveM1($\mathbf{r}$):
  Solve $\hat{\mathbf{y}}$ by preconditioned Lanczos with $(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W},\mathbf{C}^\top\mathbf{r},\mathbf{M}_2^{-1},O(\log\kappa/\epsilon_0))$; ▷ Alg. 6.
  Compute $\hat{\mathbf{w}}=\frac{1}{\tilde{\lambda}}(\mathbf{r}-\mathbf{C}\hat{\mathbf{y}})$;
return $\hat{\mathbf{w}}$; ▷ Solves $\mathbf{M}\mathbf{w}=\mathbf{r}$ for $\mathbf{M}=\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top+\tilde{\lambda}\mathbf{I}$.

4.1 Proof of Theorem 4.1 We give the proof in the following four parts.

Part 1: Nyström Preconditioner with Sparse Embedding. Let $\mathbf{S}\in\mathbb{R}^{s\times n}$ be the sparse embedding matrix according to Definition 2.1 with $s=O(l\log(l/\delta))$ and $\gamma=O(\log(l/\delta))$. Denote $\mathbf{C}\coloneqq\mathbf{A}\mathbf{S}^\top$ and $\mathbf{W}\coloneqq\mathbf{S}\mathbf{A}\mathbf{S}^\top$, then we have $\hat{\mathbf{A}}_{\text{nys}}=\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top$ and $\mathbf{M}=\hat{\mathbf{A}}_{\text{nys}}+\tilde{\lambda}\mathbf{I}$ where $\tilde{\lambda}=\lambda+\lambda_0$. According to Lemma 4.2, with probability at least $1-\delta/2$ we have $\kappa_{\mathbf{M}}\le\frac{C\bar{\kappa}_{l,\lambda}n}{l}$ for some $C=O(1)$. For the proof of Lemma 4.2 and a detailed discussion of sparse embedding matrices, see Section 4.2.

LEMMA 4.2. (PRECONDITIONER BASED ON SPARSE EMBEDDING) *Given positive definite matrix $\mathbf{A}\in\mathbb{R}^{n\times n}$ and $\delta\in(0,1/2)$. For $\log n<l<n$, let $\hat{\mathbf{A}}_{\text{nys}}=\mathbf{A}\mathbf{S}^\top(\mathbf{S}\mathbf{A}\mathbf{S}^\top)^{-1}\mathbf{S}\mathbf{A}$ be the Nyström approximation of $\mathbf{A}$, where $\mathbf{S}\in\mathbb{R}^{s\times n}$ is the sparse embedding matrix with $s=O(l\log(l/\delta))$ and $\gamma=O(\log(l/\delta))$. Given $\lambda\ge0$, let $\mathbf{M}=\hat{\mathbf{A}}_{\text{nys}}+(\lambda+\lambda_0)\mathbf{I}$ where $\lambda_0=\frac{2}{l}\sum_{i\ge l+1}\lambda_i$, then with probability $1-\delta$ the condition number $\kappa_{\mathbf{M}}$ of the matrix $\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$ satisfies:*

$$\kappa_{\mathbf{M}}\le\frac{C\bar{\kappa}_{l,\lambda}n}{l}\quad\textit{for some}\quad C=O(1).$$

Part 2: Level-1 Preconditioning. For the first level we use preconditioned Lanczos (Algorithm 6) to solve $(\mathbf{A}+\lambda\mathbf{I})\mathbf{x}=\mathbf{b}$ with preconditioner $\mathbf{M}$, where we need to compute $\mathbf{M}^{-1}\mathbf{r}$ in each iteration. The following lemma guarantees that by using inverse formula, we can approximate $\mathbf{M}^{-1}\mathbf{r}$ well enough by solving a linear system with matrix $\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}$ to a certain accuracy. For the proof of Lemma 4.3, see Section 4.3.

LEMMA 4.3. *Given matrix $\mathbf{A}\in\mathcal{S}_n^{++}$, let $\hat{\mathbf{A}}_{\text{nys}}=\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top$ be its Nyström approximation where $\mathbf{C}=\mathbf{A}\mathbf{S}^\top$ and $\mathbf{W}=\mathbf{S}\mathbf{A}\mathbf{S}^\top$. For $\tilde{\lambda}>0$, denote $\mathbf{M}=\hat{\mathbf{A}}_{\text{nys}}+\tilde{\lambda}\mathbf{I}$ as the Nyström preconditioner and assume that $\mathbf{M}\approx_2\mathbf{A}+\tilde{\lambda}\mathbf{I}$. Given vector $\mathbf{r}\in\mathbb{R}^n$ and $\epsilon_1>0$, suppose we can solve the linear system $(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})\mathbf{y}=\mathbf{C}^\top\mathbf{r}$ and compute $\hat{\mathbf{y}}$ such that*

$$\|\hat{\mathbf{y}}-\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}\le\epsilon_1\cdot\|\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}\quad\textit{where}\quad\mathbf{y}^*\coloneqq(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\mathbf{r}.$$

Then, $\hat{\mathbf{w}}=\frac{1}{\tilde{\lambda}}(\mathbf{r}-\mathbf{C}\hat{\mathbf{y}})$ is an approximate solution of the linear system $\mathbf{M}\mathbf{w}=\mathbf{r}$ satisfying

$$\|\hat{\mathbf{w}}-\mathbf{M}^{-1}\mathbf{r}\|_{\mathbf{M}}\le c_0\epsilon_1\kappa^{3/2}\|\mathbf{M}^{-1}\mathbf{r}\|_{\mathbf{M}}\quad\textit{for some}\quad c_0=O(1).\tag{4.2}$$

According to Eq.(4.5) in the proof of Lemma 4.3, we have the following holds:

$$\mathbf{M}^{-1} = \frac{1}{\tilde{\lambda}}\left(\mathbf{I} - \mathbf{C}(\mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\right).$$

By using this inverse formula, we have $\mathbf{w}^* \coloneqq \mathbf{M}^{-1}\mathbf{r} = \frac{1}{\tilde{\lambda}}(\mathbf{r} - \mathbf{C}\mathbf{y}^*)$. Instead of directly computing $\mathbf{y}^*$, suppose we can solve the linear system $(\mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W})\mathbf{y} = \mathbf{C}^\top\mathbf{r}$ approximately and have the guarantee (which will be proved in Level-2 analysis later) that $\|\hat{\mathbf{y}} - \mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}} \le \epsilon_1 \cdot \|\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}$, then Lemma 4.3 guarantees that Eq.(4.2) holds as long as $\mathbf{M} \approx_2 \mathbf{A} + \tilde{\lambda}\mathbf{I}$. The assumption here can be easily verified: for one side we have

$$\mathbf{M} = \hat{\mathbf{A}}_{\text{nys}} + \tilde{\lambda}\mathbf{I} \preceq \mathbf{A} + \tilde{\lambda}\mathbf{I} \prec 2(\mathbf{A} + \tilde{\lambda}\mathbf{I})$$

and for the other side, using that $\|\hat{\mathbf{A}}_{\text{nys}} - \mathbf{A}\| \le \lambda_0$ we have

$$2\mathbf{M} - (\mathbf{A} + \tilde{\lambda}\mathbf{I}) = 2\hat{\mathbf{A}}_{\text{nys}} - \mathbf{A} + \tilde{\lambda}\mathbf{I} \succeq \hat{\mathbf{A}}_{\text{nys}} - \mathbf{A} + \lambda_0\mathbf{I} \succeq \mathbf{0}$$

which together give $\mathbf{M} \approx_2 \mathbf{A} + \tilde{\lambda}\mathbf{I}$. Thus we can use Lemma 4.3 and have Eq.(4.2) holds. Now given any $\epsilon > 0$, if in Eq.(4.2) we set $\epsilon_1 = O(\epsilon_0/\kappa^{3/2})$ for $\epsilon_0 = O(\frac{1}{(\kappa_{\mathbf{M}}\cdot n/\epsilon)^c})$, then by Lemma 4.1, after $t_{\max} = O(\sqrt{\kappa_{\mathbf{M}}}\log(\kappa_{\mathbf{M}}/\epsilon))$ iterations, preconditioned Lanczos outputs $\tilde{\mathbf{x}}$ such that

$$\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_{\mathbf{A}+\lambda\mathbf{I}} \le \epsilon \cdot \|\mathbf{x}^*\|_{\mathbf{A}+\lambda\mathbf{I}}.$$

Thus, we obtain the final convergence result. Next, we analyze the second level.

Part 3: Level-2 Preconditioning. For the second level we use preconditioned Lanczos to solve the $s \times s$ linear system $(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})\mathbf{y} = \mathbf{C}^\top\mathbf{r}$. As for the preconditioner we choose $\mathbf{M}_2 = \tilde{\mathbf{C}}^\top\tilde{\mathbf{C}}+\tilde{\lambda}\mathbf{W} = \mathbf{C}^\top\mathbf{\Phi}^\top\mathbf{\Phi}\mathbf{C}+\tilde{\lambda}\mathbf{W}$, where $\mathbf{\Phi} \in \mathbb{R}^{\phi\times n}$ is a sparse sketching matrix constructed using standard subspace embedding techniques [CDDR24] (see Lemma 2.2) with choice $\phi = O(s + \log 1/\delta)$. By Lemma 2.2, with probability $1 - \delta/2$ we have $\kappa_{\mathbf{M}_2} \coloneqq \kappa(\mathbf{M}_2^{-1/2}(\mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W})\mathbf{M}_2^{-1/2}) = O(1)$. Since in Algorithm 1 we pre-compute $\mathbf{M}_2^{-1}$ exactly, the error caused by the preconditioner is 0. By using Theorem 6.1 we know that after $O(\sqrt{\kappa_{\mathbf{M}_2}}\log(\kappa_{\mathbf{M}_2}/\epsilon_1)) = O(\log 1/\epsilon_1) = O(\log(\kappa/\epsilon_0))$ iterations, preconditioned Lanczos will output $\hat{\mathbf{y}}$ that satisfies

$$\|\hat{\mathbf{y}} - \mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}} \le \epsilon_1 \cdot \|\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}$$

which is exactly the assumption we need in Level-1. By applying a union bound we finish the proof.

Part 4: Total Cost. Finally we consider the total cost of Algorithm 1. First we pre-compute $\mathbf{C} = \mathbf{A}\mathbf{S}^\top$ and $\mathbf{W} = \mathbf{S}\mathbf{C}$ which takes $O(\gamma\cdot\text{nnz}(\mathbf{A}) + \gamma\cdot\text{nnz}(\mathbf{C}))$ time. Notice that $\gamma = O(\log(l/\delta))$ and $\text{nnz}(\mathbf{C}) \le \gamma\cdot\text{nnz}(\mathbf{A})$, thus this step takes $O(\text{nnz}(\mathbf{A})\log^2(l/\delta))$ time. Next, we pre-compute $\tilde{\mathbf{C}} = \mathbf{\Phi}\mathbf{C}$ and $\mathbf{M}_2^{-1} = (\tilde{\mathbf{C}}^\top\tilde{\mathbf{C}} + \tilde{\lambda}\mathbf{W})^{-1}$ using Lemma 2.2, which takes $O(\text{nnz}(\mathbf{C})\log(s/\delta) + s^2\log^4(s/\delta) + s^\omega) = O(\text{nnz}(\mathbf{A})\log^2(l/\delta) + l^\omega\log^\omega(l/\delta))$. By Theorem 6.1 and above analysis, the number of iterations for Level-1 preconditioned Lanczos is $t_{\max} = O(\sqrt{\kappa_{\mathbf{M}}}\log(\kappa_{\mathbf{M}}/\epsilon)) = O(\sqrt{\bar{\kappa}_{l,\lambda}n/l}\log(\bar{\kappa}_{l,\lambda}n/\epsilon))$.

In Level-2, in each iteration we need to compute matrix-vector products of $\mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W}$. Notice that $(\mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W})\mathbf{x} = (\mathbf{S}\mathbf{A} + \mathbf{S})\mathbf{A}\mathbf{S}^\top\mathbf{x}$ for any vector $\mathbf{x}$, thus we can first compute $\mathbf{S}^\top\mathbf{x}$ which takes $O(n\log(l/\delta))$, then compute $\mathbf{A}(\mathbf{S}^\top\mathbf{x})$ which takes $O(\text{nnz}(\mathbf{A}))$, and finally apply $\mathbf{S}\mathbf{A}$ and $\mathbf{S}$ to this vector which takes $O(n\log(l/\delta) + \text{nnz}(\mathbf{A}))$ time. Since there are $O(\log 1/\epsilon_1) = O(\log(\kappa/\epsilon_0))$ iterations of preconditioned Lanczos for Level-2, we conclude that the total cost is

$$\begin{aligned}
&O\left(\text{nnz}(\mathbf{A})\log^2(l/\delta) + l^\omega\log^\omega(l/\delta) + \text{nnz}(\mathbf{A})\log(l/\delta)\log(\kappa/\epsilon_0)\cdot\sqrt{\bar{\kappa}_{l,\lambda}n/l}\log(\bar{\kappa}_{l,\lambda}n/\epsilon)\right)\\
=\ &O\left(\text{nnz}(\mathbf{A})\sqrt{\bar{\kappa}_{l,\lambda}n/l}\log^2(l/\delta)\log(\kappa/\epsilon_0)\log(\bar{\kappa}_{l,\lambda}n/\epsilon) + l^\omega\log^\omega(l/\delta)\right)\\
=\ &\tilde{O}\left(\text{nnz}(\mathbf{A})\sqrt{\bar{\kappa}_{l,\lambda}n/l}\log^2(\kappa/\epsilon) + l^\omega\right).
\end{aligned}$$

4.2 Coarse Nyström Preconditioner Based on Sparse Embedding In this section we give a proof of Lemma 4.2, which measures the quality of Nyström preconditioner based on a sparse embedding matrix. Notice that, compared to the most commonly used guarantee that the preconditioned linear system has a constant condition number, we are using a "coarse" Nyström preconditioner which benefits us by reducing the cost of constructing it, while slightly sacrificing its quality and allowing the condition number after preconditioning to be "not so small".

To characterize the properties of sparse embedding matrix, we first introduce the following notion of oblivious subspace embedding moments property, which can be seen as a generalization of the "JL-moment property" introduced by [KN14].

DEFINITION 4.1. (DEFINITION 4 IN [CNW16]) *A distribution $\mathcal{D}$ over $\mathbb{R}^{s\times n}$ has (ϵ,δ,d,ℓ)-OSE moments if for all matrices $\mathbf{U}\in\mathbb{R}^{n\times d}$ with orthonormal columns, we have*

$$\mathbb{E}_{\mathbf{\Pi}\sim\mathcal{D}}\|(\mathbf{\Pi U})^\top(\mathbf{\Pi U})-\mathbf{I}\|^\ell<\epsilon^\ell\cdot\delta.$$

[Coh16] showed that a sparse embedding matrix (Definition 2.1) with sketch size $s=O(d\log(d/\delta)/\epsilon^2)$ and $\gamma=O(\log(d/\delta)/\epsilon)$ non-zeros per column satisfies $(\epsilon,\delta,d,\log(d/\delta))$-OSE moments property, as stated in the following lemma.

LEMMA 4.4. (THEOREM 4.2 IN [COH16]) *For any $\epsilon,\delta\in(0,1/2)$ and $B>2$, a sparse subspace embedding matrix $\mathbf{S}$ with $s=O(Bd\log(d/\delta)/\epsilon^2)$ and $\gamma=O(\log_B(d/\delta)/\epsilon)$ satisfies*

$$\mathbb{E}\,\|(\mathbf{SU})^\top(\mathbf{SU})-\mathbf{I}\|^{\log(d/\delta)}<\epsilon^{\log(d/\delta)}\cdot\delta.$$

[CNW16] showed that the aforementioned OSE moment property is a sufficient condition for the approximated matrix multiplication (AMM), see Lemma 4.5. Moreover, as a corollary they provided a low rank approximation result based on the AMM property, see Lemma 4.6.

LEMMA 4.5. (THEOREM 6 IN [CNW16]) *Given $k,\epsilon,\delta\in(0,1/2)$, let $\mathcal{D}$ be any distribution over matrices with n columns with the $(\epsilon,\delta,2k,\ell)$-OSE moment property for some $\ell\geq 2$, then for any $\mathbf{A},\mathbf{B}$ we have*

$$\Pr_{\mathbf{\Pi}\sim\mathcal{D}}\left(\left\|(\mathbf{\Pi A})^\top(\mathbf{\Pi B})-\mathbf{A}^\top\mathbf{B}\right\|\geq\epsilon\sqrt{(\|\mathbf{A}\|^2+\|\mathbf{A}\|_F^2/k)(\|\mathbf{B}\|^2+\|\mathbf{B}\|_F^2/k)}\right)<\delta.$$

LEMMA 4.6. (THEOREM 8 IN [CNW16]) *For matrix $\mathbf{A}\in\mathbb{R}^{n\times d}$, let $\mathbf{A}=\mathbf{U\Sigma V}^\top$ be its SVD and $\mathbf{A}_k=\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\top$ be its best rank-k approximation. If $\mathbf{S}\in\mathbb{R}^{s\times n}$ satisfies the following two properties:*

1. *approximate spectral norm matrix multiplication w.r.t. $\mathbf{U}_k$ and $\mathbf{A}_{\bar{k}}\coloneqq\mathbf{A}-\mathbf{A}_k$, that is:*

$$\|\mathbf{U}_k^\top\mathbf{S}^\top\mathbf{S}\mathbf{A}_{\bar{k}}-\mathbf{U}_k^\top\mathbf{A}_{\bar{k}}\|^2\leq\frac{\epsilon}{8}\left(\|\mathbf{U}_k\|^2+\frac{\|\mathbf{U}_k\|_F^2}{k}\right)\left(\|\mathbf{A}_{\bar{k}}\|^2+\frac{\|\mathbf{A}_{\bar{k}}\|_F^2}{k}\right)$$

2. *$\mathbf{S}$ is a $1/2$-subspace embedding for the column space of $\mathbf{U}_k$,*

then, defining $\mathbf{P}_\mathbf{S}$ as the orthogonal projection onto the row-space of $\mathbf{SA}$, and denoting $\tilde{\mathbf{A}}_k$ as the best rank-k approximation of $\tilde{\mathbf{A}}=\mathbf{AP}_\mathbf{S}$, we have:

$$\|\mathbf{A}-\tilde{\mathbf{A}}_k\|^2\leq(1+\epsilon)\|\mathbf{A}-\mathbf{A}_k\|^2+(\epsilon/k)\cdot\|\mathbf{A}-\mathbf{A}_k\|_F^2.$$

In order to use Lemma 4.6 to get a low-rank approximation result, there are two requirements that we need to satisfy. As for the first approximate spectral norm matrix multiplication assumption, it follows directly since the OSE moment property holds; as for the second subspace embedding property, it can be shown from the definition of sparse embedding matrix. Formally, we give the proof of Lemma 4.2 as follows.

Proof. [Proof of Lemma 4.2] For positive definite matrix $\mathbf{A}$, we denote $\mathbf{Z}\coloneqq\mathbf{A}^{1/2}$ be such that $\mathbf{A}=\mathbf{ZZ}^\top$. Given $\delta<1/2$ and $l<n$, we first verify that the sparse embedding matrix $\mathbf{S}\in\mathbb{R}^{s\times n}$ with $s=O(l\log(l/\delta))$ and $\gamma=O(\log(l/\delta))$ satisfies the two requirements of Lemma 4.6 on matrix $\mathbf{Z}$:

- Requirement 1: we use Lemma 4.4 with choice $d = l$ and $\epsilon = 1/2\sqrt{2}$, thus the sparse embedding matrix with $s = O(l\log(l/\delta))$ and $\gamma = O(\log(l/\delta))$ satisfies the $(1/2\sqrt{2}, \delta/2, l, \log(l/\delta))$-OSE moment property. By using Lemma 4.5 to matrix $\mathbf{U}_k$ and $\mathbf{A}_{\bar{k}}$, we can show the first requirement holds to $\epsilon = 1$ with probability at least $1 - \delta/2$.
- Requirement 2: we also use Lemma 4.4 with Markov's inequality, which shows that $\mathbf{S}$ satisfies

$$\|(\mathbf{S}\mathbf{U}_l)^\top(\mathbf{S}\mathbf{U}_l) - \mathbf{U}_l^\top\mathbf{U}_l\| \le \frac{1}{2\sqrt{2}} < \frac{1}{2}$$

  with probability at least $1 - \delta/2$. Thus we have the $1/2$-subspace embedding property.

With the above two requirements being satisfied, we apply Lemma 4.6 with choice $\epsilon = 1$ to matrix $\mathbf{Z}$, and have the following holds with probability $1 - \delta$:

$$\|\mathbf{Z} - \tilde{\mathbf{Z}}_l\|^2 \le 2\sigma_{l+1}^2 + \frac{1}{l}\sum_{i\ge l+1}\sigma_i^2 \tag{4.3}$$

where $\{\sigma_i\}_{i=1}^n$ are the singular values of $\mathbf{Z}$ in decreasing order. Notice that here $\tilde{\mathbf{Z}} = \mathbf{Z}\mathbf{P}_\mathbf{S}$ and $\tilde{\mathbf{Z}}_l$ is the best rank-l approximation of $\tilde{\mathbf{Z}}$, thus by denoting $\mathbf{P}_l = \mathbf{V}_l\mathbf{V}_l^\top$, we have $\tilde{\mathbf{Z}}_l = \tilde{\mathbf{Z}}\mathbf{P}_l = \mathbf{Z}\mathbf{P}_\mathbf{S}\mathbf{P}_l$. Moreover, in Lemma 4.7 we show that $\|\mathbf{Z} - \tilde{\mathbf{Z}}_l\| \ge \|\mathbf{Z} - \tilde{\mathbf{Z}}\|$, by combining this result and Eq.(4.3) we have an error bound on the term $\|\mathbf{Z} - \tilde{\mathbf{Z}}\|^2 = \|\mathbf{Z} - \mathbf{Z}\mathbf{P}_\mathbf{S}\|^2$. Further notice the following:

$$\|\mathbf{Z}(\mathbf{I} - \mathbf{P}_\mathbf{S})\|^2 = \left\|\mathbf{Z}(\mathbf{I} - \mathbf{P}_\mathbf{S})\mathbf{Z}^\top\right\| = \Big\|\underbrace{\mathbf{Z}\mathbf{Z}^\top}_{\mathbf{A}} - \underbrace{\mathbf{Z}\mathbf{Z}^\top\mathbf{S}^\top(\mathbf{S}\mathbf{Z}\mathbf{Z}^\top\mathbf{S}^\top)^{-1}\mathbf{S}\mathbf{Z}\mathbf{Z}^\top}_{\hat{\mathbf{A}}_{\text{nys}}}\Big\|.$$

Thus, if we denote $\hat{\mathbf{A}}_{\text{nys}} \coloneqq \mathbf{A}\mathbf{S}^\top(\mathbf{S}\mathbf{A}\mathbf{S}^\top)^{-1}\mathbf{S}\mathbf{A}$ as the rank-s Nyström approximation of matrix $\mathbf{A}$, then we can bound the error as

$$\|(\mathbf{A} + \lambda\mathbf{I}) - (\hat{\mathbf{A}}_{\text{nys}} + \lambda\mathbf{I})\| = \|\mathbf{A} - \hat{\mathbf{A}}_{\text{nys}}\| \le 2\sigma_{l+1}^2 + \frac{1}{l}\sum_{i\ge l+1}\sigma_i^2 = 2\lambda_{l+1} + \frac{1}{l}\sum_{i\ge l+1}\lambda_i.$$

Based on this result, if we construct the preconditioner as $\mathbf{M} = \hat{\mathbf{A}}_{\text{nys}} + \lambda\mathbf{I} + \lambda_0\mathbf{I}$ where $\lambda_0 \ge 2\lambda_{2l+1} + \frac{1}{2l}\sum_{i\ge 2l+1}\lambda_i$ and $\hat{\mathbf{A}}_{\text{nys}}$ is the rank-$2s$ (instead of rank-s) Nyström approximation, then we have $\mathbf{A} + \lambda\mathbf{I} \preceq \mathbf{M} \preceq (\mathbf{A} + \lambda\mathbf{I}) + \lambda_0\mathbf{I}$, which gives $\kappa_\mathbf{M} \le 1 + \frac{\lambda_0}{\lambda_n + \lambda}$. Further notice that we can bound this term as

$$2\lambda_{2l+1} + \frac{1}{2l}\sum_{i\ge 2l+1}\lambda_i \le \frac{2}{l}\sum_{i=l+1}^{2l}\lambda_i + \frac{1}{2l}\sum_{i\ge 2l+1}\lambda_i \le \frac{2}{l}\sum_{i\ge l+1}\lambda_i,$$

thus by setting $\lambda_0 = \frac{2}{l}\sum_{i\ge l+1}\lambda_i$ we have the following holds for some $C = O(1)$:

$$\kappa_\mathbf{M} \le 1 + \frac{\lambda_0}{\lambda_n + \lambda} \le \frac{C}{l}\sum_{i\ge l+1}\frac{\lambda_i}{\lambda_n + \lambda} \le \frac{C\bar{\kappa}_{l,\lambda}n}{l}. \tag{4.4}$$

∎

LEMMA 4.7. *We have* $\mathbf{P}_\mathbf{S}\mathbf{P}_l = \mathbf{P}_l$, *which gives* $\tilde{\mathbf{Z}}_l = \mathbf{Z}\mathbf{P}_l$. *We further have*

$$\|\mathbf{Z} - \tilde{\mathbf{Z}}_l\| \ge \|\mathbf{Z} - \tilde{\mathbf{Z}}\|.$$

Proof. Let $\tilde{\mathbf{Z}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^\top$ be its SVD, then according to definition we have $\tilde{\mathbf{Z}}_l = \tilde{\mathbf{U}}_l\tilde{\mathbf{\Sigma}}_l\tilde{\mathbf{V}}_l^\top$, and we can write the projection matrix $\mathbf{P}_l$ as $\mathbf{P}_l = \tilde{\mathbf{V}}_l\tilde{\mathbf{V}}_l^\top$ where $\tilde{\mathbf{V}}_l \in \mathbb{R}^{n\times l}$. With these, we can always write $\mathbf{P}_\mathbf{S}$ as $\mathbf{P}_\mathbf{S} = \mathbf{Q}\mathbf{Q}^\top$ where $\mathbf{Q} \in \mathbb{R}^{n\times s}$ satisfies $\tilde{\mathbf{V}}_l = \mathbf{Q}\mathbf{I}_l$ for some $\mathbf{I}_l \in \mathbb{R}^{s\times l}$ satisfying $\mathbf{I}_l^\top\mathbf{I}_l = \mathbf{I}$. Thus we have

$$\mathbf{P}_\mathbf{S}\mathbf{P}_l = \mathbf{Q}\mathbf{Q}^\top\tilde{\mathbf{V}}_l\tilde{\mathbf{V}}_l^\top = \mathbf{Q}\mathbf{Q}^\top\mathbf{Q}\mathbf{I}_l\mathbf{I}_l^\top\mathbf{Q}^\top = \mathbf{Q}\mathbf{I}_l\mathbf{I}_l^\top\mathbf{Q}^\top = \tilde{\mathbf{V}}_l\tilde{\mathbf{V}}_l^\top = \mathbf{P}_l.$$

∎

4.3 Variable Transformation In this section we give a proof of Lemma 4.3.

Proof. [Proof of Lemma 4.3] We first show the relation between the two aforementioned linear systems $(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})\mathbf{y}=\mathbf{C}^\top\mathbf{r}$ and $\mathbf{M}\mathbf{w}=\mathbf{r}$. Notice that

$$\begin{aligned}
&\mathbf{M}\cdot\left(\mathbf{I}-\mathbf{C}(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\right)\\
=&\left(\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top+\tilde{\lambda}\mathbf{I}\right)\cdot\left(\mathbf{I}-\mathbf{C}(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\right)\\
=&\,\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top-\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top\mathbf{C}(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top+\tilde{\lambda}\mathbf{I}-\tilde{\lambda}\mathbf{C}(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\\
=&\,\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top-\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top+\tilde{\lambda}\mathbf{C}(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top+\tilde{\lambda}\mathbf{I}-\tilde{\lambda}\mathbf{C}(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\\
=&\,\tilde{\lambda}\mathbf{I}
\end{aligned}$$

which gives

$$\mathbf{M}^{-1}=\frac{1}{\tilde{\lambda}}\left(\mathbf{I}-\mathbf{C}(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\right). \tag{4.5}$$

Thus, if we denote $\mathbf{y}^*=(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\mathbf{r}$ as the solution of the first linear system, then the solution of the second system can be expressed as $\mathbf{w}^*=\frac{1}{\tilde{\lambda}}(\mathbf{r}-\mathbf{C}\mathbf{y}^*)$, and we further have $\tilde{\lambda}(\hat{\mathbf{w}}-\mathbf{w}^*)=\mathbf{C}(\mathbf{y}-\hat{\mathbf{y}})$. Furthermore, by assumption $\mathbf{M}=\hat{\mathbf{A}}_{\text{nys}}+\tilde{\lambda}\mathbf{I}\approx_2\mathbf{A}+\tilde{\lambda}\mathbf{I}$ we have

$$\mathbf{C}^\top\mathbf{M}\mathbf{C}\approx_2\mathbf{C}^\top(\mathbf{A}+\tilde{\lambda}\mathbf{I})\mathbf{C}=\mathbf{S}\mathbf{A}(\mathbf{A}+\tilde{\lambda}\mathbf{I})\mathbf{A}\mathbf{S}^\top.$$

Based on this observation and the assumption that $\|\hat{\mathbf{y}}-\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}\le\epsilon_1\|\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}$, we have

$$\begin{aligned}
\tilde{\lambda}^2\|\hat{\mathbf{w}}-\mathbf{w}^*\|_{\mathbf{M}}^2&=\|\mathbf{C}(\hat{\mathbf{y}}-\mathbf{y}^*)\|_{\mathbf{M}}^2=\|\hat{\mathbf{y}}-\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{M}\mathbf{C}}^2\le 2\|\hat{\mathbf{y}}-\mathbf{y}^*\|_{\mathbf{S}\mathbf{A}(\mathbf{A}+\tilde{\lambda}\mathbf{I})\mathbf{A}\mathbf{S}^\top}^2\\
&=2\|\mathbf{A}\mathbf{S}^\top(\hat{\mathbf{y}}-\mathbf{y}^*)\|_{\mathbf{A}+\tilde{\lambda}\mathbf{I}}^2\le 2\|\mathbf{A}^{1/2}\|^2\cdot\|\mathbf{A}^{1/2}\mathbf{S}^\top(\hat{\mathbf{y}}-\mathbf{y}^*)\|_{\mathbf{A}+\tilde{\lambda}\mathbf{I}}^2\\
&=2\|\mathbf{A}\|\cdot\|\hat{\mathbf{y}}-\mathbf{y}^*\|_{\mathbf{S}(\mathbf{A}^2+\tilde{\lambda}\mathbf{A})\mathbf{S}^\top}^2=2\|\mathbf{A}\|\cdot\|\hat{\mathbf{y}}-\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}^2\\
&\le 2\epsilon_1^2\|\mathbf{A}\|\cdot\|\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}^2=2\epsilon_1^2\|\mathbf{A}\|\cdot\|\mathbf{A}^{1/2}\mathbf{S}^\top\mathbf{y}^*\|_{\mathbf{A}+\tilde{\lambda}\mathbf{I}}^2\\
&\le 2\epsilon_1^2\kappa(\mathbf{A})\cdot\|\mathbf{A}\mathbf{S}^\top\mathbf{y}^*\|_{\mathbf{A}+\tilde{\lambda}\mathbf{I}}^2=2\epsilon_1^2\kappa(\mathbf{A})\cdot\|\mathbf{y}^*\|_{\mathbf{S}\mathbf{A}(\mathbf{A}+\tilde{\lambda}\mathbf{I})\mathbf{A}\mathbf{S}^\top}^2\\
&\le 4\epsilon_1^2\kappa(\mathbf{A})\cdot\|\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{M}\mathbf{C}}^2=4\epsilon_1^2\kappa(\mathbf{A})\cdot\|\mathbf{C}\mathbf{y}^*\|_{\mathbf{M}}^2.
\end{aligned}$$

Thus we have

$$\|\hat{\mathbf{w}}-\mathbf{w}^*\|_{\mathbf{M}}\le\frac{2\epsilon_1}{\tilde{\lambda}}\sqrt{\kappa(\mathbf{A})}\cdot\|\mathbf{C}\mathbf{y}^*\|_{\mathbf{M}} \tag{4.6}$$

Furthermore, since $\mathbf{r}=\mathbf{M}\mathbf{w}^*=\frac{1}{\tilde{\lambda}}\mathbf{M}(\mathbf{r}-\mathbf{C}\mathbf{y}^*)$, we have

$$\|\mathbf{r}\|_{\mathbf{M}}=\|\mathbf{M}^{1/2}\mathbf{r}\|\le\|\mathbf{M}\|\|\mathbf{M}^{-1/2}\mathbf{r}\|=\frac{1}{\tilde{\lambda}}\|\mathbf{M}\|\|\mathbf{M}^{1/2}(\mathbf{r}-\mathbf{C}\mathbf{y}^*)\|=\frac{1}{\tilde{\lambda}}\|\mathbf{M}\|\|\mathbf{r}-\mathbf{C}\mathbf{y}^*\|_{\mathbf{M}}. \tag{4.7}$$

By combining Eq.(4.6) and (4.7), we have the following holds for some constant $c_0>4$:

$$\begin{aligned}
\|\hat{\mathbf{w}}-\mathbf{w}^*\|_{\mathbf{M}}&\le\frac{2\epsilon_1\kappa^{1/2}}{\tilde{\lambda}}\|\mathbf{C}\mathbf{y}^*\|_{\mathbf{M}}\le\frac{2\epsilon_1\kappa^{1/2}}{\tilde{\lambda}}(\|\mathbf{r}-\mathbf{C}\mathbf{y}^*\|_{\mathbf{M}}+\|\mathbf{r}\|_{\mathbf{M}})\\
&\le 2\epsilon_1\kappa^{1/2}\left(\frac{\|\mathbf{M}\|}{\tilde{\lambda}}+1\right)\cdot\frac{1}{\tilde{\lambda}}\|\mathbf{r}-\mathbf{C}\mathbf{y}^*\|_{\mathbf{M}}=2\epsilon_1\kappa^{1/2}\frac{\|\mathbf{M}+\tilde{\lambda}\mathbf{I}\|}{\tilde{\lambda}}\cdot\|\hat{\mathbf{w}}\|_{\mathbf{M}}\\
&\le 4\epsilon_1\kappa^{1/2}\frac{\|\mathbf{A}+\tilde{\lambda}\mathbf{I}\|}{\tilde{\lambda}}\cdot\|\hat{\mathbf{w}}\|_{\mathbf{M}}\le c_0\epsilon_1\kappa^{3/2}\|\hat{\mathbf{w}}\|_{\mathbf{M}}
\end{aligned}$$

where $\kappa=\kappa(\mathbf{A})$. Thus we finish the proof. ∎

5 Three-Level MSP for Solving General Linear Systems

With the easier to analyze PD case being proved in Section 4, in this section we give a proof of Theorem 3.1, where given $\mathbf{A} \in \mathbb{R}^{m\times n}$ with full column rank, $\mathbf{c} \in \mathbb{R}^n$ and $\lambda \geq 0$, we want to solve $(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I})\mathbf{x} = \mathbf{c}$. As we discuss in Section 3, such a linear system is more general in the sense that it can not only recover the rectangular linear system, but also have applications to kernel ridge regression and Schatten norm estimation.

Since $\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I} \in \mathcal{S}_n^{++}$, one natural idea is to construct the Nyström preconditioner similar as before. Given sketching matrix $\mathbf{S} \in \mathbb{R}^{s\times n}$, define $\tilde{\mathbf{A}} \coloneqq \mathbf{A}\mathbf{S}^\top, \mathbf{C} \coloneqq \mathbf{A}^\top\mathbf{A}\mathbf{S}^\top = \mathbf{A}^\top\tilde{\mathbf{A}}$ and $\mathbf{W} \coloneqq \mathbf{S}\mathbf{A}^\top\mathbf{A}\mathbf{S}^\top = \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}$. Then, $\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top$ is the classical Nyström preconditioner of matrix $\mathbf{A}^\top\mathbf{A}$. Let $\{\sigma_i\}_{i=1}^n$ be the singular values of $\mathbf{A}$ in decreasing order. By using Lemma 4.2, we can again show that the matrix $\mathbf{M} \coloneqq \mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top + \tilde{\lambda}\mathbf{I}$ where $\tilde{\lambda} \coloneqq \lambda + \frac{2}{l}\sum_{i>l}\sigma_i^2$, is a useful preconditioner for this problem, and we also have the inversion formula:

$$\mathbf{M}^{-1} = \frac{1}{\tilde{\lambda}}\left(\mathbf{I} - \mathbf{C}(\mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\right).$$

However, different from the positive definite case where we can compute $\mathbf{C}$ and $\mathbf{W}$ explicitly, in this case constructing $\mathbf{C}$ and $\mathbf{W}$ takes $\tilde{O}(s\cdot\mathrm{nnz}(\mathbf{A}))$ time which is much too expensive. To avoid computing $\mathbf{C}$, we first observe that $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top + \tilde{\lambda}\mathbf{I} \approx_2 \mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I}$ (which is proved in Lemma 5.1), concluding that

$$\mathbf{W}^2 + \tilde{\lambda}\mathbf{W} = \tilde{\mathbf{A}}^\top(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top + \tilde{\lambda}\mathbf{I})\tilde{\mathbf{A}} \approx_2 \tilde{\mathbf{A}}^\top(\mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I})\tilde{\mathbf{A}} = \mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W}.$$

With this observation, we do not need to compute $\mathbf{C}$, and we can use $\mathbf{M}_2 \coloneqq \mathbf{W}^2 + \tilde{\lambda}\mathbf{W}$ as the Level-2 preconditioner for solving the linear system $(\mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W})\mathbf{y} = \mathbf{r}$. However we still cannot afford to compute $\mathbf{W}$, not to mention $\mathbf{M}_2$. To address this, notice that

$$\begin{aligned}\mathbf{M}_2^{-1} = (\mathbf{W}^2 + \tilde{\lambda}\mathbf{W})^{-1} &= (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{-1}\\ &= (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}\frac{1}{\tilde{\lambda}}\left(\mathbf{I} - \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{-1}\right)\\ &= \frac{1}{\tilde{\lambda}}\left((\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1} - (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{-1}\right).\end{aligned}$$

Since we only need to have access to $\mathbf{M}_2^{-1}\mathbf{r}$ in Level-2 preconditioned Lanczos, it suffices to compute $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}\mathbf{r}$ and $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \lambda\mathbf{I})^{-1}\mathbf{r}$. To solve this, we introduce the third level, where we again use preconditioned Lanczos to solve two positive definite linear systems $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{u} = \mathbf{r}$ and $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})\mathbf{v} = \mathbf{r}$. To construct the Level-3 preconditioners, notice that $\tilde{\mathbf{A}}$ is an $m \times s$ tall matrix, and we can thus construct an $O(s) \times s$ sketch of it.

Let $\mathbf{\Phi} \in \mathbb{R}^{\phi\times m}$ be the oblivious embedding matrix in Lemma 2.2 with $\phi = O(s)$. We first compute $\hat{\mathbf{A}} = \mathbf{\Phi}\tilde{\mathbf{A}} \in \mathbb{R}^{\phi\times s}$, then construct the preconditioner $\mathbf{M}_{3a}^{-1} \coloneqq \hat{\mathbf{A}}^\top\hat{\mathbf{A}}$ and $\mathbf{M}_{3b}^{-1} \coloneqq \hat{\mathbf{A}}^\top\hat{\mathbf{A}} + \tilde{\lambda}\mathbf{I}$. Lemma 2.2 guarantees that both preconditioned linear systems have a constant condition number, thus solving them using preconditioned Lanczos only takes $O(\log 1/\epsilon_2)$ steps for the targeted accurancy ϵ_2. Moreover, since both $\mathbf{M}_{3a}$ and $\mathbf{M}_{3b}$ are $s \times s$ positive definite matrices, we can afford to compute their inverse exactly, and the errors caused by the Level-3 preconditioners are 0. By going back from Level-3 $\to$ Level-2 $\to$ Level-1 and analyzing the error bounds carefully, we can finally arrive at the convergence result on the original linear system.

Our algorithms are as follows. Algorithm 3 is the main algorithm for solving the linear system $(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})\mathbf{x} = \mathbf{c}$, in which we run preconditioned Lanczos with Level-1 preconditioner $\mathbf{M}$ defined by function SolverM1. Algorithm 4 defines function SolverM1 by solving linear system $\mathbf{M}\mathbf{w} = \mathbf{r}$ with preconditioned Lanczos, where the Level-2 preconditioner $\mathbf{M}_2$ is defined by function SolverM2. Lastly, Algorithm 5 defines function SolveM2 by solving two linear systems $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{u} = \mathbf{r}$ and $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})\mathbf{v} = \mathbf{r}$ with preconditioned Lanczos respectively, whereas the Level-3 preconditioners $\mathbf{M}_{3a}$ and $\mathbf{M}_{3b}$ are pre-computed (see line 6 of Algorithm 3). As a comparison of the quality of the preconditioners we use in different levels, we have:

$$\begin{cases}\text{Level-1: } \kappa_{\mathbf{M}} \coloneqq \kappa(\mathbf{M}^{-1/2}(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I})\mathbf{M}^{-1/2}) = O(\bar{\kappa}_{l,\lambda}^2 n/l) & \textbf{Coarse}\\ \text{Level-2: } \kappa_{\mathbf{M}_2} \coloneqq \kappa(\mathbf{M}_2^{-1/2}(\mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W})\mathbf{M}_2^{-1/2}) = O(1) & \textbf{Fine}\\ \text{Level-3a: } \kappa_{\mathbf{M}_{3a}} \coloneqq \kappa(\mathbf{M}_{3a}^{-1/2}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{M}_{3a}^{-1/2}) = O(1) & \textbf{Fine}\\ \text{Level-3b: } \kappa_{\mathbf{M}_{3b}} \coloneqq \kappa(\mathbf{M}_{3b}^{-1/2}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})\mathbf{M}_{3b}^{-1/2}) = O(1) & \textbf{Fine}\end{cases}$$

Algorithm 3 MSP for solving linear system $(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})\mathbf{x}=\mathbf{c}$.

Input: matrix $\mathbf{A}\in\mathbb{R}^{m\times n}$, regularize term λ and λ_0, vector $\mathbf{c}\in\mathbb{R}^n$, sparse sketch size s, # of non-zeros γ, level-2 sketch size ϕ, # of iterations $t_{\max}$;
Compute $\tilde{\lambda}=\lambda+\lambda_0$;
Construct sparse embedding matrix $\mathbf{S}\in\mathbb{R}^{s\times n}$ with γ non-zeros per column; ▷ Def. 2.1.
Compute $\tilde{\mathbf{A}}=\mathbf{A}\mathbf{S}^\top$ and $\hat{\mathbf{A}}=\mathbf{\Phi}\tilde{\mathbf{A}}\in\mathbb{R}^{\phi\times s}$; ▷ Lemma 2.2.
Compute $\mathbf{M}_{3a}=\hat{\mathbf{A}}^\top\hat{\mathbf{A}}$ and $\mathbf{M}_{3b}=\mathbf{M}_{3a}+\tilde{\lambda}\mathbf{I}$
Compute $\mathbf{M}_{3a}^{-1}$ and $\mathbf{M}_{3b}^{-1}$;
Solve $\tilde{\mathbf{x}}$ by preconditioned Lanczos with $(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I},\mathbf{c},\mathsf{SolveM1},t_{\max})$; ▷ Alg. 6.
return $\tilde{\mathbf{x}}$; ▷ Solves $(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})\mathbf{x}=\mathbf{c}$.

Algorithm 4 Level-1 auxiliary function SolveM1 for solving $\mathbf{M}\mathbf{w}=\mathbf{r}$.

function SolveM1($\mathbf{r}$):
  Solve $\hat{\mathbf{y}}$ by Lanczos with $(\tilde{\mathbf{A}}^\top\mathbf{A}\mathbf{A}^\top\tilde{\mathbf{A}}+\tilde{\lambda}\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}},\tilde{\mathbf{A}}^\top\mathbf{A}\mathbf{r},\mathsf{SolveM2},O(\log\kappa/\epsilon_0))$; ▷ Alg. 6.
  Compute $\hat{\mathbf{w}}=\frac{1}{\tilde{\lambda}}(\mathbf{r}-\mathbf{A}^\top\tilde{\mathbf{A}}\hat{\mathbf{y}})$;
return $\hat{\mathbf{w}}$; ▷ Solves $\mathbf{M}\mathbf{w}=\mathbf{r}$ for $\mathbf{M}=\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top+\tilde{\lambda}\mathbf{I}$.

Algorithm 5 Level-2 auxiliary function SolveM2 for solving $\mathbf{M}_2\mathbf{z}=\mathbf{r}$.

function SolveM2($\mathbf{r}$):
  Solve $\hat{\mathbf{u}}$ by preconditioned Lanczos with $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}},\mathbf{r},\mathbf{M}_{3a}^{-1},O(\log\kappa l/\epsilon_0))$; ▷ Alg. 6.
  Solve $\hat{\mathbf{v}}$ by preconditioned Lanczos with $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}+\tilde{\lambda}\mathbf{I},\mathbf{r},\mathbf{M}_{3b}^{-1},O(\log\kappa l/\epsilon_0))$; ▷ Alg. 6.
  Compute $\hat{\mathbf{z}}=\frac{1}{\tilde{\lambda}}(\hat{\mathbf{u}}-\hat{\mathbf{v}})$;
return $\hat{\mathbf{z}}$; ▷ Solves $\mathbf{M}_2\mathbf{z}=\mathbf{r}$ for $\mathbf{M}_2=\mathbf{W}^2+\tilde{\lambda}\mathbf{W}$.

5.1 Proof of Theorem 3.1 Given matrix $\mathbf{A}\in\mathbb{R}^{m\times n}$ and $\lambda\geq 0$, we consider to solve the regularized linear system $(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})\mathbf{x}=\mathbf{c}$ without explicitly computing $\mathbf{A}^\top\mathbf{A}$. Since $\mathbf{A}$ has full column rank, we know $\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I}\in\mathbb{R}^{n\times n}$ is positive definite. We prove the result in following five parts.

Part 1: Nyström Preconditioner with Sparse Embedding. Let $\mathbf{S}\in\mathbb{R}^{s\times n}$ be the sparse embedding matrix according to Lemma 4.2 with $s=O(l\log(l/\delta))$ and $\gamma=O(\log(l/\delta))$. Denote $\mathbf{C}:=\mathbf{A}^\top\mathbf{A}\mathbf{S}^\top$ and $\mathbf{W}:=\mathbf{S}\mathbf{A}^\top\mathbf{A}\mathbf{S}^\top$, then $\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top$ is the Nyström approximation of $\mathbf{A}^\top\mathbf{A}$. Denote $\tilde{\lambda}=\lambda+\lambda_0$, let $\mathbf{M}=\mathbf{C}\mathbf{W}^{-1}\mathbf{C}^\top+\tilde{\lambda}\mathbf{I}$ be the preconditioner, then according to Eq.(4.5) in the proof of Lemma 4.3 we have

$$\mathbf{M}^{-1}=\frac{1}{\tilde{\lambda}}\left(\mathbf{I}-\mathbf{C}(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\right).$$

Given $0<\delta<1/8$, by applying Lemma 4.2 to matrix $\mathbf{A}^\top\mathbf{A}$, if we set $\lambda_0=\frac{2}{l}\sum_{i\geq l+1}\lambda_i(\mathbf{A}^\top\mathbf{A})=\frac{2}{l}\sum_{i\geq l+1}\sigma_i^2$, then with probability at least $1-\delta/3$ we have $\kappa_{\mathbf{M}}\leq\frac{C\bar{\kappa}_{l,\lambda}^2 n}{l}$ for some $C=O(1)$.

Part 2: Level-1 Preconditioning. Similar with the positive definite case, for the first level we use preconditioned Lanczos method (Algorithm 6) where in each iteration we need to compute $\mathbf{w}^*:=\mathbf{M}^{-1}\mathbf{r}$. Similarly, we introduce $\mathbf{y}^*:=(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})^{-1}\mathbf{C}^\top\mathbf{r}$ and approximate $\mathbf{w}^*$ by $\hat{\mathbf{w}}=\frac{1}{\tilde{\lambda}}(\mathbf{r}-\mathbf{C}\hat{\mathbf{y}})$. Suppose we can solve the linear system $(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})\mathbf{y}=\mathbf{C}^\top\mathbf{r}$ and have the guarantee that (which will be proved in Level-2 and Level-3):

$$\|\hat{\mathbf{y}}-\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}\leq\epsilon_1\cdot\|\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}. \tag{5.1}$$

Notice that since $\kappa(\mathbf{A}^\top\mathbf{A})=\kappa^2$, Lemma 4.3 guarantees that

$$\|\hat{\mathbf{w}}-\mathbf{M}^{-1}\mathbf{r}\|_{\mathbf{M}}\leq c_0\epsilon_1\kappa^3\|\mathbf{M}^{-1}\mathbf{r}\|_{\mathbf{M}}\quad\text{for some}\quad c_0=O(1).$$

By setting $\epsilon_1=O(\epsilon_0/\kappa^3)$ we verify the assumption of Lemma 4.1 stated in Eq.(4.1), also note that $\kappa(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})\leq\kappa^2$. Thus by using Lemma 4.1, after $t_{\max}=O(\sqrt{\kappa_{\mathbf{M}}}\log(\kappa_{\mathbf{M}}/\epsilon))$ iterations of preconditioned Lanczos, we can obtain $\tilde{\mathbf{x}}$ such that

$$\|\tilde{\mathbf{x}}-\mathbf{x}^*\|_{\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I}}\leq\epsilon\cdot\|\mathbf{x}^*\|_{\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I}}.$$

Thus we obtain the final convergence result we want. Next, we analyze the second level.

Part 3: Level-2 Preconditioning. Compared with the positive definite case in Section 4, the difference here is that we cannot afford to compute $\mathbf{C}$ and $\mathbf{W}$, which makes the analysis more difficult. To address this, we first show in the following Lemma 5.1 that the sketch $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{S}^\top$ satisfies $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top + \tilde{\lambda}\mathbf{I} \approx_2 \mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I}$ holds with probability $1-\delta/3$.

LEMMA 5.1. (SPECTRAL APPROXIMATION) *Given matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with full column rank, $\lambda \geq 0$ and $0 < \delta < 1/2$. Given $\log n < l < n$, let $\lambda_0 = \frac{2}{l}\sum_{i>l}\sigma_i^2$ and denote $\tilde{\lambda} = \lambda + \lambda_0$. Let $\mathbf{S} \in \mathbb{R}^{n\times s}$ be the sparse subspace embedding matrix with $s = O(l\log(l/\delta))$ and each column having $\gamma = O(\log(l/\delta))$ non-zeros. Denote $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{S}^\top$, then with probability at least $1-\delta$ we have*

$$\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top + \tilde{\lambda}\mathbf{I} \approx_2 \mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I}.$$

Conditioned on this event, we have

$$\mathbf{W}^2 + \tilde{\lambda}\mathbf{W} = \tilde{\mathbf{A}}^\top(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top + \tilde{\lambda}\mathbf{I})\tilde{\mathbf{A}} \approx_2 \tilde{\mathbf{A}}^\top(\mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I})\tilde{\mathbf{A}} = \mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W}.$$

With this result, to solve the linear system $(\mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W})\mathbf{y} = \mathbf{C}^\top\mathbf{r}$, we can use preconditioned Lanczos with $\mathbf{M}_2 \coloneqq \mathbf{W}^2 + \tilde{\lambda}\mathbf{W}$ as the preconditioner, and have $\kappa_{\mathbf{M}_2} = \kappa(\mathbf{M}_2^{-1/2}(\mathbf{C}^\top\mathbf{C} + \tilde{\lambda}\mathbf{W})\mathbf{M}_2^{-1/2}) \leq 4$. However, we still can not construct $\mathbf{M}_2$ since we cannot afford to compute $\mathbf{W}$. Instead of constructing $\mathbf{M}_2$ explicitly, in each iteration we only need access to $\mathbf{M}_2^{-1}\mathbf{r}$ for a given vector $\mathbf{r}$. Further notice that $\mathbf{M}_2^{-1}$ can be expressed as follows:

$$\begin{aligned}(\mathbf{W}^2 + \tilde{\lambda}\mathbf{W})^{-1} &= (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{-1} \\ &= (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}\frac{1}{\tilde{\lambda}}\left(\mathbf{I} - \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{-1}\right) \\ &= \frac{1}{\tilde{\lambda}}\left((\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1} - (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{-1}\right),\end{aligned}$$

thus in order to compute $\mathbf{M}_2^{-1}\mathbf{r}$, it suffices to compute $\mathbf{u}^* \coloneqq (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})^{-1}\mathbf{r}$ and $\mathbf{v}^* \coloneqq (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{-1}\mathbf{r}$. Suppose we can solve two linear systems $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{u} = \mathbf{r}$, $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})\mathbf{v} = \mathbf{r}$ approximately, and have the following guarantee (which will be proved in Level-3):

$$\begin{cases}\|\hat{\mathbf{u}} - \mathbf{u}^*\|_{\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}} \leq \epsilon_{2,1}\cdot\|\mathbf{u}^*\|_{\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}}; \\ \|\hat{\mathbf{v}} - \mathbf{v}^*\|_{\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}+\tilde{\lambda}\mathbf{I}} \leq \epsilon_{2,2}\cdot\|\mathbf{v}^*\|_{\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}+\tilde{\lambda}\mathbf{I}}.\end{cases}$$

Denote $\mathbf{z}^* \coloneqq \mathbf{M}_2^{-1}\mathbf{r} = \frac{1}{\tilde{\lambda}}(\mathbf{u}^* - \mathbf{v}^*)$ as the quantity we want to approximate, and denote $\hat{\mathbf{z}} \coloneqq \frac{1}{\tilde{\lambda}}(\hat{\mathbf{u}} - \hat{\mathbf{v}})$. Observe that $\mathbf{W}^2 + \tilde{\lambda}\mathbf{W} = \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} = \tilde{\mathbf{A}}^\top(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top + \tilde{\lambda}\mathbf{I})\tilde{\mathbf{A}}$, then we have

$$\begin{aligned}\tilde{\lambda}\|\hat{\mathbf{z}} - \mathbf{z}^*\|_{\mathbf{W}^2+\tilde{\lambda}\mathbf{W}} &= \|(\hat{\mathbf{u}} - \hat{\mathbf{v}}) - (\mathbf{u}^* - \mathbf{v}^*)\|_{\mathbf{W}^2+\tilde{\lambda}\mathbf{W}} \\ &\leq \|\hat{\mathbf{u}} - \mathbf{u}^*\|_{\mathbf{W}^2+\tilde{\lambda}\mathbf{W}} + \|\hat{\mathbf{v}} - \mathbf{v}^*\|_{\mathbf{W}^2+\tilde{\lambda}\mathbf{W}} \\ &= \left\|\tilde{\mathbf{A}}(\hat{\mathbf{u}} - \mathbf{u}^*)\right\|_{\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I}} + \left\|\tilde{\mathbf{A}}(\hat{\mathbf{v}} - \mathbf{v}^*)\right\|_{\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I}} \\ &\leq \|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top + \tilde{\lambda}\mathbf{I}\|^{1/2}\cdot\left\|\tilde{\mathbf{A}}(\hat{\mathbf{u}} - \mathbf{u}^*)\right\| + \left\|\tilde{\mathbf{A}}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{1/2}(\hat{\mathbf{v}} - \mathbf{v}^*)\right\| \\ &\leq \|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top + \tilde{\lambda}\mathbf{I}\|^{1/2}\cdot\epsilon_{2,1}\|\tilde{\mathbf{A}}\mathbf{u}^*\| + \|\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}\|^{1/2}\cdot\epsilon_{2,2}\left\|(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{1/2}\mathbf{v}^*\right\|\end{aligned}$$

where in the fourth step we use the commutable property, i.e., $(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top + \tilde{\lambda}\mathbf{I})^{1/2}\tilde{\mathbf{A}} = \tilde{\mathbf{A}}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{1/2}$. Notice that from the definition of $\mathbf{u}^*$ and $\mathbf{v}^*$ we naturally have $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{u}^* = \mathbf{r} = (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})\mathbf{v}^*$, which gives $\mathbf{u}^* = \frac{1}{\tilde{\lambda}}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})(\mathbf{u}^* - \mathbf{v}^*) = (\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} + \tilde{\lambda}\mathbf{I})\mathbf{z}^*$, and also $\mathbf{v}^* = \frac{1}{\tilde{\lambda}}\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}(\mathbf{u}^* - \mathbf{v}^*) = \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}\mathbf{z}^*$. By using these

facts we can express $\mathbf{u}^*$ and $\mathbf{v}^*$ in terms of $\mathbf{z}^*$, and have

$$\begin{aligned}
&\tilde{\lambda}\|\hat{\mathbf{z}}-\mathbf{z}^*\|_{\mathbf{W}^2+\tilde{\lambda}\mathbf{W}}\\
&\le \epsilon_{2,1}\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I}\|^{1/2}\cdot\|\tilde{\mathbf{A}}\mathbf{u}^*\|+\epsilon_{2,2}\|\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}\|^{1/2}\cdot\|(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}+\tilde{\lambda}\mathbf{I})^{1/2}\mathbf{v}^*\|\\
&= \epsilon_{2,1}\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I}\|^{1/2}\cdot\|\tilde{\mathbf{A}}(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}+\tilde{\lambda}\mathbf{I})\mathbf{z}^*\|+\epsilon_{2,2}\|\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}\|^{1/2}\cdot\|(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}+\tilde{\lambda}\mathbf{I})^{1/2}\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}\mathbf{z}^*\|\\
&= \epsilon_{2,1}\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I}\|^{1/2}\cdot\|(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I})\tilde{\mathbf{A}}\mathbf{z}^*\|+\epsilon_{2,2}\|\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}\|^{1/2}\cdot\|\tilde{\mathbf{A}}^\top(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I})^{1/2}\tilde{\mathbf{A}}\mathbf{z}^*\|\\
&\le \epsilon_{2,1}\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I}\|\cdot\|(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I})^{1/2}\tilde{\mathbf{A}}\mathbf{z}^*\|+\epsilon_{2,2}\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top\|\cdot\|(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I})^{1/2}\tilde{\mathbf{A}}\mathbf{z}^*\|\\
&= \left(\epsilon_{2,1}\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I}\|+\epsilon_{2,2}\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top\|\right)\cdot\|\mathbf{z}^*\|_{\mathbf{W}^2+\tilde{\lambda}\mathbf{W}}.
\end{aligned}$$

Thus given $\epsilon_0>0$, by choosing $\epsilon_{2,1}\le\frac{\epsilon_0\tilde{\lambda}}{2\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I}\|}$ and $\epsilon_{2,2}\le\frac{\epsilon_0\tilde{\lambda}}{2\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top\|}$, we have

$$\|\hat{\mathbf{z}}-\mathbf{z}^*\|_{\mathbf{W}^2+\tilde{\lambda}\mathbf{W}}\le\epsilon_0\cdot\|\mathbf{z}^*\|_{\mathbf{W}^2+\tilde{\lambda}\mathbf{W}}. \tag{5.2}$$

Notice that $\mathbf{z}^*=\mathbf{M}_2^{-1}\mathbf{r}$. With this result, we apply Lemma 4.1 to linear system $(\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W})\mathbf{y}=\mathbf{C}^\top\mathbf{r}$ with preconditioner $\mathbf{M}_2$, and obtain $\hat{\mathbf{y}}$ such that

$$\|\hat{\mathbf{y}}-\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}\le\epsilon_1\cdot\|\mathbf{y}^*\|_{\mathbf{C}^\top\mathbf{C}+\tilde{\lambda}\mathbf{W}}$$

in $O(\sqrt{\kappa_{\mathbf{M}_2}}\log(\kappa_{\mathbf{M}_2}/\epsilon_1))=O(\log 1/\epsilon_1)$ iterations, since $\kappa_{\mathbf{M}_2}=O(1)$. Notice that this is exactly the requirement we need for Level-1 (see Eq.(5.1)). Moreover, we have the following bounds:

$$\begin{aligned}
\frac{\epsilon_0\tilde{\lambda}}{2\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top+\tilde{\lambda}\mathbf{I}\|}&\ge\frac{\epsilon_0\tilde{\lambda}}{4\|\mathbf{A}\mathbf{A}^\top+\tilde{\lambda}\mathbf{I}\|}\ge\frac{\epsilon_0\lambda_0}{4(\sigma_1^2+\lambda_0)}=\frac{\epsilon_0\lambda_0/\sigma_n^2}{4\kappa^2+4\lambda_0/\sigma_n^2}\\
&=\frac{\epsilon_0(n-l)\bar{\kappa}_l^2}{2\kappa^2 l+4(n-l)\bar{\kappa}_l^2}\ge\frac{\epsilon_0(n-l)}{2\kappa^2 l+4(n-l)}\ge\frac{\epsilon_0}{2\kappa^2 l+4}\ge\frac{\epsilon_0}{4\kappa^2 l}
\end{aligned}$$

where we use the fact that $\frac{\lambda_0}{\sigma_n^2}=\frac{2(n-l)}{l}\bar{\kappa}_l^2$ and $\bar{\kappa}_l\ge 1$, and similarly,

$$\frac{\epsilon_0\tilde{\lambda}}{2\|\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top\|}\ge\frac{\epsilon_0\tilde{\lambda}}{4\|\mathbf{A}\mathbf{A}^\top\|}\ge\frac{\epsilon_0\lambda_0}{4\|\mathbf{A}\mathbf{A}^\top\|}=\frac{\epsilon_0\lambda_0}{4\sigma_1^2}\ge\frac{\epsilon_0(n-l)\bar{\kappa}_{l,\lambda}^2}{2\kappa^2 l}\ge\frac{\epsilon_0}{4\kappa^2 l}.$$

Thus we can choose $\epsilon_{2,1}=\epsilon_{2,2}=\epsilon_0/4\kappa^2 l$ to guarantee that the requirement we need for Level-2 (see Eq.(5.2)) holds. Finally we step into the innermost Level-3.

Part 4: Level-3 Preconditioning. As the innermost level (Algorithm 5), we use preconditioned Lanczos to solve two positive definite linear systems $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}})\mathbf{u}=\mathbf{r}$ and $(\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}+\tilde{\lambda}\mathbf{I})\mathbf{v}=\mathbf{r}$, with preconditioners $\mathbf{M}_{3a}:=\hat{\mathbf{A}}^\top\hat{\mathbf{A}}$ and $\mathbf{M}_{3b}:=\hat{\mathbf{A}}^\top\hat{\mathbf{A}}+\tilde{\lambda}\mathbf{I}$ respectively, where $\hat{\mathbf{A}}:=\mathbf{\Phi}\tilde{\mathbf{A}}\in\mathbb{R}^{\phi\times s}$ and $\mathbf{\Phi}$ is the oblivious embedding matrix defined by Lemma 2.2 with $\phi=O(s+\log(1/\delta))$. With probability $1-\delta/3$ we have $\kappa_{\mathbf{M}_{3a}}=\kappa_{\mathbf{M}_{3b}}=O(1)$. Since in Algorithm 3 we pre-compute $\mathbf{M}_{3a}^{-1}$ and $\mathbf{M}_{3b}^{-1}$ exactly, the error terms for these two preconditioners are 0. By using Lemma 4.1, we need $O(\sqrt{\kappa_{\mathbf{M}_{3a}}}\log(\kappa_{\mathbf{M}_{3a}}/\epsilon_{2,1}))$ and $O(\sqrt{\kappa_{\mathbf{M}_{3b}}}\log(\kappa_{\mathbf{M}_{3b}}/\epsilon_{2,2}))$ iterations respectively to guarantee

$$\begin{cases}\|\hat{\mathbf{u}}-\mathbf{u}^*\|_{\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}}\le\epsilon_{2,1}\cdot\|\mathbf{u}^*\|_{\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}};\\ \|\hat{\mathbf{v}}-\mathbf{v}^*\|_{\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}+\tilde{\lambda}\mathbf{I}}\le\epsilon_{2,2}\cdot\|\mathbf{v}^*\|_{\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}+\tilde{\lambda}\mathbf{I}}.\end{cases}$$

Furthermore, since we choose $\epsilon_{2,1}=\epsilon_{2,2}=\epsilon_0/4\kappa^2 l$, thus the number of iterations in Level-3 are $O(\sqrt{\kappa_{\mathbf{M}_{3a}}}\log(\kappa_{\mathbf{M}_{3a}}/\epsilon_{2,1}))=O(\log 1/\epsilon_{2,1})=O(\log(\kappa l/\epsilon_0))$. Notice that the above guarantees are exactly the requirements we need for Level-2, thus, by going back from Level-3 $\to$ Level-2 $\to$ Level-1 and applying a union bound over the probabilities of "$\kappa_{\mathbf{M}}=O(\bar{\kappa}_{l,\lambda}^2 n/l),\kappa_{\mathbf{M}_2}=\kappa_{\mathbf{M}_{3a}}=\kappa_{\mathbf{M}_{3b}}=O(1)$", we conclude that $\|\tilde{\mathbf{x}}-\mathbf{x}^*\|_{\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I}}\le\epsilon\|\mathbf{x}^*\|_{\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I}}$ holds with probability $1-\delta$.

Part 5: Total Cost. Finally we consider the total cost of Algorithm 3 together with Algorithm 4 and 5. We first pre-compute $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{S}^\top$ which takes $O(\mathrm{nnz}(\mathbf{A})\log(l/\delta))$, and also compute $\hat{\mathbf{A}} = \mathbf{\Phi}\tilde{\mathbf{A}}$ which takes $O(\mathrm{nnz}(\tilde{\mathbf{A}})\log(s/\delta) + s^2\log^4(s/\delta))$ according to Lemma 2.2. With these, we can compute the Level-3 preconditioners $\mathbf{M}_{3a}^{-1} = (\hat{\mathbf{A}}^\top\hat{\mathbf{A}})^{-1}$ and $\mathbf{M}_{3b}^{-1} = (\hat{\mathbf{A}}^\top\hat{\mathbf{A}} + \tilde{\lambda}\mathbf{I})^{-1}$ in time $O(s^\omega)$. To conclude, line 2-6 of Algorithm 3 takes in total:

$$O(\mathrm{nnz}(\tilde{\mathbf{A}})\log(s/\delta) + s^2\log^4(s/\delta) + s^\omega) = O(\mathrm{nnz}(\mathbf{A})\log^2(l/\delta) + l^\omega\log^\omega(l/\delta)).$$

Within each iteration, we need to compute matrix-vector products of matrix $\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I}$ which takes $O(\mathrm{nnz}(\mathbf{A}))$, and call SolveM1 as in Algorithm 4. Denote $T_{\mathbf{M}^{-1}}$ as the time of calling SolveM1, and denote $T_{\mathbf{M}_2^{-1}}$ as the time of calling SolveM2 as in Algorithm 5. Observe that in each iteration of Algorithm 5 we need to compute $\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}\mathbf{r} = \mathbf{S}\mathbf{A}^\top\mathbf{A}\mathbf{S}^\top\mathbf{r}$ for any given vector $\mathbf{r}$, which takes $O(n\log(l/\delta) + \mathrm{nnz}(\mathbf{A})) = O(\mathrm{nnz}(\mathbf{A})\log(l/\delta))$. Since there are $O(\log(\kappa l/\epsilon_0))$ iterations, the complexity for Algorithm 5 becomes

$$T_{\mathbf{M}_2^{-1}} = O\left(\mathrm{nnz}(\mathbf{A})\log(l/\delta)\log(\kappa l/\epsilon_0)\right).$$

In each iteration of Algorithm 4 we need to compute $\tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}\mathbf{r}$ which is similar, thus we have

$$\begin{aligned} T_{\mathbf{M}^{-1}} &= O((\mathrm{nnz}(\mathbf{A})\log(l/\delta) + T_{\mathbf{M}_2^{-1}})\cdot\log 1/\epsilon_1) \\ &= O(\mathrm{nnz}(\mathbf{A})\log(l/\delta)\log(\kappa l/\epsilon_0)\log 1/\epsilon_1) \\ &= O(\mathrm{nnz}(\mathbf{A})\log(l/\delta)\log(\kappa l/\epsilon_0)\log(\kappa/\epsilon_0)). \end{aligned}$$

Finally, since there are $t_{\max} = O(\bar{\kappa}_{l,\lambda}\sqrt{n/l}\cdot\log(\bar{\kappa}_{l,\lambda}n/\epsilon))$ iterations in total, we compute the overall complexity of Algorithm 3 as the following:

$$\begin{aligned} &O(\underbrace{\mathrm{nnz}(\mathbf{A})\log^2(l/\delta) + l^\omega\log^\omega(l/\delta)}_{\text{preconditioning}}) + \underbrace{(\mathrm{nnz}(\mathbf{A}) + T_{\mathbf{M}^{-1}})\cdot\bar{\kappa}_{l,\lambda}\sqrt{n/l}\log(\bar{\kappa}_{l,\lambda}n/\epsilon))}_{\text{iteration}} \\ &= O(\mathrm{nnz}(\mathbf{A})\log^2(l/\delta) + l^\omega\log^\omega(l/\delta) + \mathrm{nnz}(\mathbf{A})\log(l/\delta)\log(\kappa l/\epsilon_0)\log(\kappa/\epsilon_0)\cdot\bar{\kappa}_{l,\lambda}\sqrt{n/l}\log(\bar{\kappa}_{l,\lambda}n/\epsilon)) \\ &= O(\mathrm{nnz}(\mathbf{A})\log^2(l/\delta)\log^2(\kappa/\epsilon_0)\cdot\bar{\kappa}_{l,\lambda}\sqrt{n/l}\log(\bar{\kappa}_{l,\lambda}n/\epsilon) + l^\omega\log^\omega(l/\delta)) \\ &= \tilde{O}\left(\mathrm{nnz}(\mathbf{A})\sqrt{\frac{n}{l}}\bar{\kappa}_{l,\lambda}\log^3(\kappa/\epsilon) + l^\omega\right). \end{aligned}$$

5.2 Proof of Lemma 5.1 For $\tilde{\lambda} = \lambda + \lambda_0 \geq \frac{2}{l}\sum_{i>l}\sigma_i^2$, denote $\mathbf{\Sigma}_\lambda = \mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I}$ and $\mathbf{B} = \mathbf{A}^\top\mathbf{\Sigma}_{\tilde{\lambda}}^{-1/2}$. Here we list some basic properties of $\mathbf{B}$:

$$\begin{cases} \|\mathbf{B}\|^2 = \|\mathbf{B}\mathbf{B}^\top\| = \|\mathbf{A}^\top\mathbf{\Sigma}_{\tilde{\lambda}}^{-1}\mathbf{A}\| \leq 1 \\ \|\mathbf{B}\|_F^2 = \mathrm{tr}(\mathbf{B}\mathbf{B}^\top) = \mathrm{tr}(\mathbf{A}\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I})^{-1}) =: d_{\tilde{\lambda}} \end{cases}$$

By using Lemma 4.4, we know that the sparse subspace embedding matrix with $s = O(d_{\tilde{\lambda}}\log(d_{\tilde{\lambda}}/\delta))$ and each column having $\gamma = O(\log(d_{\tilde{\lambda}}/\delta))$ satisfies the $(\frac{1}{6}, \delta, d_{\tilde{\lambda}}, \log(d_{\tilde{\lambda}}/\delta))$-OSE moment property, thus by using Lemma 4.5, with probability $1-\delta$ we have

$$\|\mathbf{B}^\top\mathbf{S}^\top\mathbf{S}\mathbf{B} - \mathbf{B}^\top\mathbf{B}\| \leq \frac{1}{6}(\|\mathbf{B}\|^2 + 2\|\mathbf{B}\|_F^2/d_{\tilde{\lambda}}) \leq \frac{1}{2}.$$

Conditioned on this guarantee, we have the following holds:

$$\begin{aligned} &\mathbf{B}^\top\mathbf{B} - \frac{1}{2}\mathbf{I} \preceq \mathbf{B}^\top\mathbf{S}^\top\mathbf{S}\mathbf{B} \preceq \mathbf{B}^\top\mathbf{B} + \frac{1}{2}\mathbf{I} \\ \Leftrightarrow\ &\mathbf{\Sigma}_{\tilde{\lambda}}^{-1/2}\mathbf{A}\mathbf{A}^\top\mathbf{\Sigma}_{\tilde{\lambda}}^{-1/2} - \frac{1}{2}\mathbf{I} \preceq \mathbf{\Sigma}_{\tilde{\lambda}}^{-1/2}\mathbf{A}\mathbf{S}^\top\mathbf{S}\mathbf{A}^\top\mathbf{\Sigma}_{\tilde{\lambda}}^{-1/2} \preceq \mathbf{\Sigma}_{\tilde{\lambda}}^{-1/2}\mathbf{A}\mathbf{A}^\top\mathbf{\Sigma}_{\tilde{\lambda}}^{-1/2} + \frac{1}{2}\mathbf{I} \\ \Leftrightarrow\ &\mathbf{A}\mathbf{A}^\top - \frac{1}{2}(\mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I}) \preceq \mathbf{A}\mathbf{S}^\top\mathbf{S}\mathbf{A}^\top \preceq \mathbf{A}\mathbf{A}^\top + \frac{1}{2}(\mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I}) \\ \Leftrightarrow\ &\frac{1}{2}(\mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I}) \preceq \mathbf{A}\mathbf{S}^\top\mathbf{S}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I} \preceq \frac{3}{2}(\mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I}) \\ \Rightarrow\ &\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top + \tilde{\lambda}\mathbf{I} \approx_2 \mathbf{A}\mathbf{A}^\top + \tilde{\lambda}\mathbf{I}. \end{aligned}$$

Finally we bound $d_{\tilde{\lambda}}$ by l by using the fact that $\tilde{\lambda} \geq \frac{2}{l}\sum_{i>l}\sigma_i^2$:

$$d_{\tilde{\lambda}} = \sum_{i=1}^{n}\frac{\sigma_i^2}{\sigma_i^2+\tilde{\lambda}} = \sum_{i=1}^{l}\frac{\sigma_i^2}{\sigma_i^2+\tilde{\lambda}} + \sum_{i=l+1}^{n}\frac{\sigma_i^2}{\sigma_i^2+\tilde{\lambda}} \leq l + \sum_{i=l+1}^{n}\frac{\sigma_i^2 l}{2\sum_{j>l}\sigma_j^2} = \frac{3}{2}l,$$

thus we conclude that $s = O(l\log(l/\delta))$ and $\gamma = O(\log(l/\delta))$ suffices for the spectral approximation.

6 Analysis of Inexact Preconditioned Lanczos Iteration

Our Multi-level Sketched Preconditioning (MSP) algorithms described in Sections 4 and 5 rely on multiple calls to a preconditioned Lanczos algorithm. This method requires routines for matrix-vector multiplication with $\mathbf{A}$, as well as matrix-vector multiplication with the inverse of the preconditioner, $\mathbf{M}$. A key feature of our framework is that, in many cases, we do not explicitly construct $\mathbf{M}^{-1}$, but rather compute $\mathbf{M}^{-1}\mathbf{r}$ for a given vector $\mathbf{r} \in \mathbb{R}^n$ *approximately*. It is not clear a priori that the outer iterative method used in our algorithms is robust to this approximation, i.e. that Lanczos still converges as quickly as the case when $\mathbf{M}^{-1}$ is applied exactly. In particular, while there has been significant work showing that the *unpreconditioned* Lanczos method is robust to inexact matrix-vector multiplications, and actually to round-off error in all arithmetic operations [Pai76, Gre89, MMS18], there is less work on analyzing *preconditioned* Lanczos.

One line of work studies the closely related Preconditioned Conjugate Gradient (PCG) method, which returns exactly the same output as Lanczos when all computations are performed exactly [GY99]. That work establishes that PCG still converges when $\mathbf{M}^{-1}$ is applied approximately. However, it has the disadvantage of only providing a *non-accelerated* convergence rate. I.e., it establishes convergence depending on the condition number $\kappa_{\mathbf{M}} = \kappa(\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2})$ of the preconditioned system. On the other hand, if $\mathbf{M}^{-1}$ is applied exactly, a dependence on $\sqrt{\kappa_{\mathbf{M}}}$ is achievable. This better $\sqrt{\kappa_{\mathbf{M}}}$ dependence is critical for obtaining our final results. Other efforts to analyze the stability of iterative methods also fail to obtain an accelerated rate. For example, there has been interesting recent work on the stability of preconditioned gradient descent [Epp24], but any gradient descent type method will have convergence depending on $\kappa_{\mathbf{M}}$ instead of $\sqrt{\kappa_{\mathbf{M}}}$.

We address this issue by proving a new result for the Lanczos method with inexact preconditioning that maintains the accelerated rate. We suspect that a similar result can be achieved for PCG, which is more commonly used in practice than Lanczos, although we leave this effort to future work. As discussed in Section 1.2, we also note that our analysis of Lanczos is not strictly necessary for this paper. For example, [ST14b] give an analysis of inexact preconditioned *Chebyshev iteration*, which could have been used instead. However, the preconditioned Lanczos method is easier to implement (e.g., does not require eigenvalue bounds) and much more widely used in practice since it tends to converge much more quickly than Chebyshev iteration, even though it has the same worst-case guarantees [ACG+23]. As such, we expect that our stability analysis of the preconditioned Lanczos method will be of general interest beyond this work.

6.1 Inexact Preconditioned Lanczos Method

Concretely, in this section we analyze the preconditioned Lanczos method described in Algorithm 6. The specific implementation of Lanczos in Algorithm 6 is based on an implementation of an unpreconditioned Lanczos method whose numerical stability was analyzed in seminal work by Christopher Paige [Pai71, Pai76]. Paige's work was later used to show that Lanczos can stably solve linear systems and more general matrix function problems on finite precision computers [Gre89, MMS18]. Our analysis will rely on these previous results. Formally, we prove the following result:

THEOREM 6.1. *Consider solving* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *for positive definite* $\mathbf{A}$ *using Algorithm 6 provided with a function* SolveM *that, for some positive semidefinite preconditioner* $\mathbf{M}$ *and any vector* $\mathbf{r}$ *returns*

$$\|\mathsf{SolveM}(\mathbf{r}) - \mathbf{M}^{-1}\mathbf{r}\|_{\mathbf{M}} \leq \epsilon_0 \cdot \|\mathbf{M}^{-1}\mathbf{r}\|_{\mathbf{M}}. \tag{6.1}$$

If $\epsilon_0 \leq (\frac{\epsilon}{\kappa_{\mathbf{M}} n})^c$ *for a fixed constant* $c > 0$*, where* $\kappa_{\mathbf{M}}$ *is the condition number of* $\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$*, then the algorithm outputs* $\mathbf{x}_t$ *such that* $\|\mathbf{x}_t - \mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}} \leq \epsilon\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}$ *in* $t = O(\sqrt{\kappa_{\mathbf{M}}}\log(\kappa_{\mathbf{M}}/\epsilon))$ *iterations.*

We prove Theorem 6.1 in an indirect way by comparing the output of Algorithm 6 to the output of a more easily analyzed variant of Lanczos with *symmetric preconditioning*, which is given as Algorithm 7. This variant

Algorithm 6 Preconditioned Lanczos Iteration

Input: positive definite $\mathbf{A} \in \mathbb{R}^{n \times n}$, vector $\mathbf{b} \in \mathbb{R}^n$, function SolveM, # of iterations t;
Output: vector $\mathbf{x}_t \in \mathbb{R}^n$ that approximates $\mathbf{A}^{-1}\mathbf{b}$;
$\overline{\mathbf{w}_0} \leftarrow \mathsf{SolveM}(\mathbf{b}), z' = \sqrt{\langle \mathbf{b}, \overline{\mathbf{w}_0} \rangle}$;
$\underline{\mathbf{q}_0} = \mathbf{0}, \overline{\mathbf{q}_1} = \mathbf{w}'/z, \underline{\mathbf{q}_1} = \mathbf{b}/z, \beta'_1 = 0$;
for $i = 1, \ldots, t$ **do**
  $\underline{\mathbf{u}_i} \leftarrow \mathbf{A}\overline{\mathbf{q}_i} - \beta'_i \underline{\mathbf{q}_{i-1}}$;
  $\alpha'_i \leftarrow \langle \underline{\mathbf{u}_i}, \overline{\mathbf{q}_i} \rangle$;
  $\underline{\mathbf{w}_i} \leftarrow \underline{\mathbf{u}_i} - \alpha'_i \underline{\mathbf{q}_i}$, $\overline{\mathbf{w}_i} = \mathsf{SolveM}(\underline{\mathbf{w}_i})$;
  $\beta'_{i+1} \leftarrow \langle \underline{\mathbf{w}_i}, \overline{\mathbf{w}_i} \rangle^{1/2}$;
  if $\beta'_{i+1} == 0$ **then**
    break;
  end if
  $\underline{\mathbf{q}_{i+1}} \leftarrow \underline{\mathbf{w}_i}/\beta'_{i+1}$, $\overline{\mathbf{q}_{i+1}} \leftarrow \overline{\mathbf{w}_i}/\beta'_{i+1}$;
end for
$\mathbf{T}' \in \mathbb{R}^{t \times t} \leftarrow \begin{bmatrix} \alpha'_1 & \beta'_2 & & 0 \\ \beta'_2 & \alpha'_2 & \ddots & \\ & \ddots & \ddots & \beta'_t \\ 0 & & \beta'_t & \alpha'_t \end{bmatrix}$, $\overline{\mathbf{Q}} \in \mathbb{R}^{n \times t} \leftarrow \begin{bmatrix} \overline{\mathbf{q}_1} & \ldots & \overline{\mathbf{q}_t} \end{bmatrix}$;
return $\mathbf{x}'_t = z' \overline{\mathbf{Q}} \mathbf{T}'^{-1} \mathbf{e}_1$ where $\mathbf{e}_1 \in \mathbb{R}^t$ denotes the first standard basis vector.

Algorithm 7 Symmetric Preconditioned Lanczos Iteration.

Input: positive definite $\mathbf{A} \in \mathbb{R}^{n \times n}$, vector $\mathbf{b} \in \mathbb{R}^n$, positive definite preconditioner $\mathbf{M} \in \mathbb{R}^{n \times n}$, # of iterations t;
Output: vector $\mathbf{y} \in \mathbb{R}^n$ that approximates $\mathbf{A}^{-1}\mathbf{b}$;
$\mathbf{w}_0 \leftarrow \mathbf{M}^{-1/2}\mathbf{b}$, $z \leftarrow \|\mathbf{w}_0\|$;
$\mathbf{q}_0 \leftarrow \mathbf{0}, \mathbf{q}_1 \leftarrow \mathbf{w}/z, \beta_1 = 0$;
for $i = 1, \ldots, t$ **do**
  $\mathbf{u}_i \leftarrow \mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}\mathbf{q}_i - \beta_i \mathbf{q}_{i-1}$;
  $\alpha_i \leftarrow \langle \mathbf{u}_i, \mathbf{q}_i \rangle$;
  $\mathbf{w}_i \leftarrow \mathbf{u}_i - \alpha_i \mathbf{q}_i$;
  $\beta_{i+1} \leftarrow \|\mathbf{w}_i\|$;
  if $\beta_{i+1} == 0$ **then**
    break;
  end if
  $\mathbf{q}_{i+1} \leftarrow \mathbf{w}_i/\beta_{i+1}$;
end for
$\mathbf{T} \in \mathbb{R}^{t \times t} \leftarrow \begin{bmatrix} \alpha_1 & \beta_2 & & 0 \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ 0 & & \beta_k & \alpha_k \end{bmatrix}$, $\mathbf{Q} \in \mathbb{R}^{n \times t} \leftarrow \begin{bmatrix} \mathbf{q}_1 & \ldots & \mathbf{q}_k \end{bmatrix}$;
return $\mathbf{x}_t = \mathbf{M}^{-1/2}(z\mathbf{Q}\mathbf{T}^{-1}\mathbf{e}_1)$, where $\mathbf{e}_1 \in \mathbb{R}^t$ denotes the first standard basis vector.

assumes access to the symmetrically preconditioned matrix $\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$. To motivate the algorithm, observe that:

$$\mathbf{A}^{-1}\mathbf{b} = \mathbf{M}^{-1/2}(\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2})^{-1}\mathbf{M}^{-1/2}\mathbf{b}.$$

Algorithm 7 is obtained by just running the standard unpreconditioned Lanczos method (see e.g., [MMS18]) with right hand side $\mathbf{M}^{-1/2}\mathbf{b}$ and matrix $(\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2})$, and then multiplying the final result by $\mathbf{M}^{-1/2}$.

In our application, we only have access to $\mathbf{M}^{-1}$, so we cannot implement Algorithm 7 directly: it is only used for our analysis of Algorithm 6. In particular, we will take advantage of the fact that it can be easily shown that Algorithm 7 is robust to implementation in *finite precision arithmetic*. This is because, unlike Algorithm 6, Algorithm 7 is formally equivalent to unpreconditioned Lanczos run with a particular choice of inputs. The robustness of the *unpreconditioned* Lanczos method to round-off errors has been widely studied.

Formally, suppose all basic arithmetic operators are computed up to *relative accuracy*. I.e., for any operation $x \circ y$ where $\circ \in \{+, -, \times, \div\}$, the computer running Algorithm 7 returns a result

$$z = (1+\delta)(x \circ y) \qquad \text{such that} \qquad |\delta| \leq \epsilon_{\text{mach}}, \tag{6.2}$$

where ϵ_{mach} is the machine precision. Similarly, for any x, the computer can compute $z = (1+\delta)(x \circ y)$ where $|\delta| \leq \epsilon_{\text{mach}}$. In this setting, we have the following bound:

FACT 6.1. (COROLLARY OF THEOREM 1 [MMS18]) *Suppose Algorithm 7 is given access to the exact value of* $\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$ *and* $\mathbf{M}^{-1/2}\mathbf{b}$ *and is then implemented on a finite precision computer with machine precision* $\epsilon_{\text{mach}} = \text{poly}(\epsilon/n\kappa_{\mathbf{M}})$*, except that the final multiplication by* $\mathbf{M}^{-1/2}$ *on line 16 is performed exactly. Here* $\kappa_{\mathbf{M}} = \kappa(\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2})$*. Then the algorithm will return* $\mathbf{x}_t$ *such that* $\|\mathbf{x}_t - \mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}} \leq \epsilon\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}$ *after* $t = O\left(\sqrt{\kappa_{\mathbf{M}}}\log(\kappa_{\mathbf{M}}/\epsilon)\right)$ *iterations.*

Proof. This fact is obtained by applying Theorem 1 of [MMS18] to solving the positive definite preconditioned system $\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}\mathbf{x} = \mathbf{M}^{-1/2}\mathbf{b}$, which has solution $\mathbf{M}^{1/2}\mathbf{A}^{-1}\mathbf{b}$. Observe that Algorithm 7 is exactly equivalent to the Lanczos method analyzed in that work, except for the last step: to return an approximate solution to $\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}\mathbf{x} = \mathbf{M}^{-1/2}\mathbf{b}$, we should return just

$$\mathbf{y} = z\mathbf{Q}\mathbf{T}^{-1}\mathbf{e}_1$$

instead of $\mathbf{x}_t = \mathbf{M}^{-1/2}(z\mathbf{Q}\mathbf{T}^{-1}\mathbf{e}_1)$. Let $\lambda_1 \geq \ldots \geq \lambda_n > 0$ denote the eigenvalues of the $\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$. We apply Theorem 1 with $f(x) = 1/x$ and $\eta = \lambda_n/2$, and use the well known fact (see, e.g. [She94]) that for any $\delta < 1$, there exists a degree $O(\sqrt{\kappa_{\mathbf{M}}}\log(1/\delta))$ polynomial such that:

$$\max_{x \in [\lambda_n, \lambda_1]} |p(x) - 1/x| \leq \frac{\delta}{\lambda_n}.$$

Setting $\delta = \text{poly}(\epsilon/\kappa_{\mathbf{M}})$ and $\epsilon_{\text{mach}} = \text{poly}(n\kappa_{\mathbf{M}}/\epsilon)$, Theorem 1 from [MMS18] implies that after $t = O(\sqrt{\kappa_{\mathbf{M}}}\log(\kappa_{\mathbf{M}}/\epsilon))$ iterations,

$$\|\mathbf{y} - \mathbf{M}^{1/2}\mathbf{A}^{-1}\mathbf{b}\| \leq \frac{\epsilon}{\kappa_{\mathbf{M}}\lambda_n}\|\mathbf{M}^{-1/2}\mathbf{b}\|. \tag{6.3}$$

Using that $\mathbf{x}_t = \mathbf{M}^{-1/2}\mathbf{y}$ exactly (since we assumed the last multiplication by $\mathbf{M}^{-1/2}$ is perfomed exactly on Line 16, we have:

$$\begin{aligned}\|\mathbf{y} - \mathbf{M}^{1/2}\mathbf{A}^{-1}\mathbf{b}\| &= \|(\mathbf{A}^{1/2}\mathbf{M}^{-1/2})^{-1}\mathbf{A}^{1/2}\mathbf{M}^{-1/2}\mathbf{y} - (\mathbf{A}^{1/2}M^{-1/2})^{-1}\mathbf{A}^{-1/2}\mathbf{b}\| \\ &\geq \frac{1}{\sqrt{\lambda_1}}\|\mathbf{A}^{1/2}\mathbf{x}_t - \mathbf{A}^{-1/2}\mathbf{b}\| = \frac{1}{\sqrt{\lambda_1}}\|\mathbf{x}_t - \mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}.\end{aligned}$$

We further have that:

$$\frac{\epsilon}{\kappa_{\mathbf{M}}\lambda_n}\|\mathbf{M}^{-1/2}\mathbf{b}\| = \frac{\epsilon}{\kappa_{\mathbf{M}}\lambda_n}\|(\mathbf{A}^{-1/2}\mathbf{M}^{1/2})^{-1}\mathbf{A}^{-1/2}\mathbf{b}\| \leq \frac{\epsilon}{\kappa_{\mathbf{M}}\lambda_n}\sqrt{\lambda_1}\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}.$$

Plugging both of these bounds into (6.3), we conclude that, as desired.

$$\|\mathbf{x}_t - \mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}} \leq \frac{\epsilon\lambda_1}{\kappa_{\mathbf{M}}\lambda_n}\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}} = \epsilon\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}.$$

∎

We can actually slightly strengthen Fact 6.1. In particular, the analysis in [MMS18] is based in a black-box way on a bound from [Pai76] on the output of the general Lanczos tridiagonalization method in finite precision. This result is stated as Theorem 8 in [MMS18] and is the first theorem in [Pai76]. So, the fact actually holds for *any sequence of round-off errors* for which this critical theorem remains true.

FACT 6.2. *The result of Fact 6.1 holds when Algorithm 7 is implemented with any sequence of round-off errors for which the main Theorem of [Pai76] (Theorem 8 in [MMS18]) holds with* $\epsilon_{\text{mach}} = \text{poly}(n\kappa_{\mathbf{M}}/\epsilon)$. *This includes, e.g., any sequence of adversarially chosen round-off errors on basic arithmetic operations that satisfy (6.2), even if those are not precisely the errors that would be made in a standard implementation of finite precision arithmetic.*

We leverage Fact 6.2 to prove Theorem 6.1 by arguing that the output of Algorithm 6, the preconditioned Lanczos method that we hope to analyze, is *exactly equivalent* to the output of Algorithm 7 for some choice of round-off errors that satisfy the requirements to prove Paige's theorem. The proof is completed below. It is not self-contained: the reader will require access to [Pai76]. As of April 2024, a copy can be found on the author's webpage.

Proof. [Proof of Theorem 6.1] Before beginning the proof, we make a remark on notation. The variable names in Algorithm 6 are chosen to mirror the variable names in Algorithm 7. In particular, if Algorithm 6 and Algorithm 7 are implemented in exact arithmetic and with exact applications of all matrix-vector multiplications, it can be checked that any variable $\mathbf{x}$ satisfies the relationship

$$\overline{\mathbf{x}} = \mathbf{M}^{-1/2}\mathbf{x} \qquad \text{and} \qquad \underline{\mathbf{x}} = \mathbf{M}^{1/2}\mathbf{x}.$$

E.g., at Line 6 of Algorithm 6, $\underline{\mathbf{u}_i}$ would exactly equal $\mathbf{M}^{1/2}\mathbf{u}_i$ computed at Line 6 of Algorithm 7. Additionally, any variable that appears with a single tick mark in Algorithm 6 would be exactly equal to the corresponding variable in Algorithm 7. I.e. we would have that $z' = z$ on Line 3.

To motivate Algorithm 6, note that it is essentially equivalent to Algorithm 7, except that, instead of keeping track of $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$, it keeps track of $\mathbf{M}^{-1/2}\mathbf{Q} = [\overline{\mathbf{q}_1}, \ldots, \overline{\mathbf{q}_k}]$. This can be done without every having access to $\mathbf{M}^{-1/2}$ and, importantly, allows use to avoid the final step of Algorithm 7, which requires multiplying $\mathbf{M}^{-1/2}$ by a vector in the span of $\mathbf{Q}$.

Concretely, we will prove that there is a sequence of round-off errors in Algorithm 7 for which Paige's main theorem holds, and for which $\mathbf{T}' = \mathbf{T}$ and $\overline{\mathbf{Q}} = \mathbf{M}^{-1/2}\mathbf{Q}$, where $\mathbf{T}'$ and $\overline{\mathbf{Q}}$ are the quantities computed by Algorithm 6 run with a procedure SolveM for applying $\mathbf{M}^{-1}$ that satisfies the accuracy guarantee of (6.1). We will actually argue that this equivalence holds when Algorithm 7 is run on a slight perturbation of $\mathbf{b}$, $\mathbf{b}'$. We will account for this difference towards the end of the proof.

Our proof will proceed via induction. In particular, we will prove that for all i,

$$\alpha_i' = \alpha_i \qquad \beta_i' = \beta_i, \qquad \overline{\mathbf{q}_i} = \mathbf{M}^{-1/2}\mathbf{q}_i \qquad \text{and} \qquad \|\underline{\mathbf{q}_i} - \mathbf{M}^{1/2}\mathbf{q}_i\|_{\mathbf{M}^{-1}} \leq \epsilon_{\text{mach}}\,. \tag{6.4}$$

To establish this bound, we will inductively assume that it holds for all $j \leq i$. We begin with the base cases, which includes any variable set outside of the for loop on Line 5.

Base Case. We start by noting that trivially, $\mathbf{0} = \underline{\mathbf{q}_0} = \mathbf{M}^{1/2}\mathbf{q}_0 = \mathbf{0}$ and $0 = \beta_1' = \beta_1 = 0$. Moreover, by (6.1), $\overline{\mathbf{w}_0} = \mathbf{M}^{-1}\mathbf{b} + \Delta_0$, where Δ_0 is a vector with $\|\Delta_0\|_{\mathbf{M}} \leq \epsilon_0\|\mathbf{M}^{-1}\mathbf{b}\|_{\mathbf{M}}$. I.e. $\overline{\mathbf{w}_0} = \mathbf{M}^{-1}(\mathbf{b} + \mathbf{M}\Delta_0)$. So, for some $\mathbf{b}'$ satisfying

$$\mathbf{b}' - \mathbf{b} = \mathbf{M}\Delta_0, \tag{6.5}$$

we have that, for $\mathbf{w}_0 = \mathbf{M}^{-1/2}\mathbf{b}'$,

$$\overline{\mathbf{w}_0} = \mathbf{M}^{-1/2}\mathbf{w}_0. \tag{6.6}$$

Next, we have that

$$(z')^2 = \langle \mathbf{b}, \overline{\mathbf{w}_0}\rangle = \|\mathbf{M}^{-1/2}\mathbf{b}\|^2 + \langle \Delta_0, \mathbf{b}\rangle.$$

By Cauchy-Schwarz, $|\langle \Delta_0, \mathbf{b}\rangle| \leq \|\mathbf{M}^{-1/2}\mathbf{b}\|\|\mathbf{M}^{1/2}\Delta_0\|$ and we have that $\|\mathbf{M}^{1/2}\Delta_0\| = \|\Delta_0\|_{\mathbf{M}} \leq \epsilon_0\|\mathbf{M}^{-1/2}\mathbf{b}\|_2$. It follows that, as long as $\epsilon_0 \leq \epsilon_{\text{mach}}$, $(z')^2 = (1+\Delta_1)\|\mathbf{M}^{-1/2}\mathbf{b}\|^2$ for some scalar Δ_1 with $|\Delta_1| \leq \epsilon_{\text{mach}}$, and therefore that:

$$z' = (1+\Delta_2)\|\mathbf{M}^{-1/2}\mathbf{b}\|$$

for scalar Δ_2 with $|\Delta_2| \leq \epsilon_{\text{mach}}$. On Line 3 of Algorithm 7, Paige's analysis allows for $z \leftarrow \|\mathbf{w}_0\|$ to be computed up to relative error $(1+\epsilon_{\text{mach}}(n+2)/2)$ (see [Pai76], equation 12). So, there is a choice of acceptable roundoff error for which $z' = z$. Combined with (6.6), we conclude that:

$$\overline{\mathbf{q}_1} = \mathbf{M}^{-1/2}\mathbf{q}_1. \tag{6.7}$$

This establishes the second equation in (6.4), so to complete the base case analysis for $i = 1$, we are left to address the third equation, i.e. that $\|\underline{\mathbf{q}_1} - \mathbf{M}^{1/2}\mathbf{q}_1\| \leq \epsilon_{\text{mach}}\|\mathbf{M}^{1/2}\|$. We have from (6.7) that $\mathbf{M}^{1/2}\mathbf{q}_1 = \mathbf{M}\overline{\mathbf{q}_1} = \mathbf{M}\cdot(\mathbf{M}^{-1}\mathbf{b} + \Delta_0)/z'$. We have that $\underline{\mathbf{q}_1} = \mathbf{b}/z'$, so we conclude that:

$$\|\underline{\mathbf{q}_1} - \mathbf{M}^{1/2}\mathbf{q}_1\|_{\mathbf{M}^{-1}} = \|\mathbf{M}\Delta_0/z'\|_{\mathbf{M}^{-1}} = \|\Delta_0/z'\|_{\mathbf{M}}.$$

As shown above, as long as $\epsilon_{\text{mach}} \leq 1/2$, $z' \geq \frac{1}{2}\|\mathbf{M}^{-1/2}\mathbf{b}\|$, so we have that

$$\|\underline{\mathbf{q}_1} - \mathbf{M}^{1/2}\mathbf{q}_1\|_{\mathbf{M}^{-1}} \leq \frac{2}{\|\mathbf{M}^{-1/2}\mathbf{b}\|}\|\Delta_0\|_{\mathbf{M}} \leq 2\epsilon_0.$$

So, we have prove the third equation of (6.4) as long as $\epsilon_0 \leq \frac{\epsilon_{\text{mach}}}{2}$.

Inductive Case. We can now move onto the inductive case of (6.4). I.e., to proving the statement for all $i \geq 2$, assuming it holds for $j < i$.

We proceed with a line by line analysis of Algorithms 6 and 7.

Line 6. Our first goal is to prove that for some choice of acceptable roundoff error in Algorithm 7,

$$\mathbf{M}^{-1/2}\underline{\mathbf{u}_i} = \mathbf{u}_i. \tag{6.8}$$

Observe that

$$\begin{aligned}
\mathbf{M}^{-1/2}\underline{\mathbf{u}_i} &= \mathbf{M}^{-1/2}\mathbf{A}\overline{\mathbf{q}_i} - \beta_i'\mathbf{M}^{-1/2}\underline{\mathbf{q}_{i-1}} \\
&= \mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}\mathbf{q}_i - \beta_i\mathbf{M}^{-1/2}\underline{\mathbf{q}_{i-1}} \\
&= \mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}\mathbf{q}_i - \beta_i\mathbf{q}_{i-1} + \beta_i\Delta_3,
\end{aligned}$$

where, using our inductive assumption, $\Delta_3 = \mathbf{q}_{i-1} - \mathbf{M}^{-1/2}\underline{\mathbf{q}_{i-1}}$ is a vector with Euclidean norm bounded by ϵ_{mach}. Paige's analysis allows for $\mathbf{u}_i \leftarrow \mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}\mathbf{q}_i - \beta_i\mathbf{q}_{i-1}$ to be computed up to additive error bounded in norm by $\epsilon_{\text{mach}}\, 2\|\beta_i\mathbf{q}_{i-1}\|$ (see [Pai76], equation 8).

Using that $\|\mathbf{q}_i\| \geq 1/2$ as long as $\epsilon_{\text{mach}} \leq \frac{1}{2(n+4)}$ (see [Pai76] equation 14) we thus have that there is a choice of allowable error in Algorithm 7 such that $\mathbf{M}^{-1/2}\underline{\mathbf{u}_i} = \mathbf{u}_i$. So, we have proven (6.8) as desired.

Line 7. Next, by our inductive assumption and (6.9), we have that

$$\langle \underline{\mathbf{u}_i}, \overline{\mathbf{q}_i}\rangle = \langle \mathbf{u}_i, \mathbf{q}_i\rangle,$$

so we immediately have that, even with no roundoff error in Algorithm 7,

$$\alpha_i' = \alpha_i. \tag{6.9}$$

This establishes the first condition of (6.4).

Line 8. We want to prove that, for some choice of acceptable rounding error in Algorithm 7,

$$\mathbf{M}^{1/2}\overline{\mathbf{w}_i} = \mathbf{w}_i. \tag{6.10}$$

By Equation (6.9) and Equation (6.1) we have:

$$\overline{\mathbf{w}_i} = \mathbf{M}^{-1}\underline{\mathbf{u}_i} - \alpha_i \mathbf{M}^{-1}\underline{\mathbf{q}_i} + \Delta_4, \tag{6.11}$$

where $\|\Delta_4\|_{\mathbf{M}} \leq \epsilon_0 \|\mathbf{M}^{-1/2}(\underline{\mathbf{u}_i} - \alpha_i \underline{\mathbf{q}_i})\|$. Furthermore, by (6.4), we have that $\mathbf{M}^{-1/2}\underline{\mathbf{q}_i} = \mathbf{q}_i + \Delta_5$, where $\|\Delta_5\| \leq \epsilon_{\text{mach}}$. Combining with (6.8), it follows that:

$$\mathbf{M}^{1/2}\overline{\mathbf{w}_i} = \mathbf{M}^{-1/2}\underline{\mathbf{u}_i} - \alpha_i \mathbf{M}^{-1/2}\underline{\mathbf{q}_i} + \mathbf{M}^{1/2}\Delta_4 \tag{6.12}$$

$$= \mathbf{u}_i - \alpha_i \mathbf{q}_{i+1} + \alpha_i \Delta_5 + \mathbf{M}^{1/2}\Delta_4. \tag{6.13}$$

Paige's analysis requires that $\mathbf{w}_i$ equals $\mathbf{u}_i - \alpha_i \mathbf{q}_i$ up to additive error with norm bounded by $\epsilon_{\text{mach}}(\|\mathbf{u}_i\| + 2\alpha_i \|\mathbf{q}_i\|)$ (see [Pai76], equation 8). So to establish (6.10), we need to show that:

$$\|\alpha_i \Delta_5 + \mathbf{M}^{1/2}\Delta_4\| \leq \epsilon_{\text{mach}}(\|\mathbf{u}_i\| + 2\alpha_i \|\mathbf{q}_i\|). \tag{6.14}$$

Observe that, by triangle inequality,

$$\begin{aligned}\|\mathbf{M}^{1/2}\Delta_4\| \leq \epsilon_0 \|\mathbf{M}^{-1/2}(\underline{\mathbf{u}_i} - \alpha_i \underline{\mathbf{q}_i})\| &= \epsilon_0 \|\mathbf{u}_i - \alpha_i \mathbf{M}^{-1/2}\underline{\mathbf{q}_i}\| \\ &\leq \epsilon_0 \|\mathbf{u}_i - \alpha_i \mathbf{q}_i\| + \alpha_i \Delta_5.\end{aligned}$$

Combined with our bound on Δ_5, we have:

$$\|\alpha_i \Delta_5 + \mathbf{M}^{1/2}\Delta_4\| \leq \epsilon_0 \|\mathbf{u}_i\| + 2\alpha_i\, \epsilon_{\text{mach}}\,.$$

Using that $\|\mathbf{q}_i\| \in [1/2, 2]$ as long as $\epsilon_{\text{mach}} \leq \frac{1}{2(n+4)}$ (see [Pai76] equation 14), we see that (6.14) holds as long as $\epsilon_0 \leq \epsilon_{\text{mach}}$. It follows that (6.10) holds.

We also claim that, directly from (6.10) and (6.1),

$$\|\mathbf{w}_i - \mathbf{M}^{-1/2}\underline{\mathbf{w}_i}\| = \|\overline{\mathbf{w}_i} - \mathbf{M}^{-1}\underline{\mathbf{w}_i}\|_{\mathbf{M}} \leq \epsilon_0 \|\mathbf{M}^{-1}\underline{\mathbf{w}_i}\|_{\mathbf{M}} = \epsilon_0 \|\mathbf{M}^{-1/2}\underline{\mathbf{w}_i}\|. \tag{6.15}$$

Line 9. By (6.15) and (6.10), we have that:

$$\langle \underline{\mathbf{w}_i}, \overline{\mathbf{w}_i} \rangle = \langle \mathbf{M}^{-1/2}\underline{\mathbf{w}_i}, \mathbf{M}^{1/2}\overline{\mathbf{w}_i} \rangle = \langle \mathbf{w}_i, \mathbf{w}_i \rangle + \langle \Delta_6, \mathbf{w}_i \rangle,$$

where $\|\Delta_6\| \leq \epsilon_0 \|\mathbf{M}^{-1/2}\underline{\mathbf{w}_i}\|$. Applying Cauchy-Schwarz, it follows that:

$$\langle \underline{\mathbf{w}_i}, \overline{\mathbf{w}_i} \rangle = \langle \mathbf{w}_i, \mathbf{w}_i \rangle + \Delta_7,$$

where Δ_7 is a scalar with $|\Delta_7| \leq \epsilon_0 \|\mathbf{w}_i\| \|\mathbf{M}^{-1/2}\underline{\mathbf{w}_i}\|$. As long as $\epsilon_0 \leq 1/2$, we can check from (6.15) that $\|\mathbf{M}^{-1/2}\underline{\mathbf{w}_i}\| \leq 2\|\mathbf{w}_i\|$. It follows that $\langle \underline{\mathbf{w}_i}, \overline{\mathbf{w}_i} \rangle = (1 + \Delta_8)\langle \mathbf{w}_i, \mathbf{w}_i \rangle$ where $|\Delta_8| \leq 2\epsilon_0$, and thus

$$\sqrt{\langle \underline{\mathbf{w}_i}, \overline{\mathbf{w}_i} \rangle} = (1 + \Delta_9)\sqrt{\langle \mathbf{w}_i, \mathbf{w}_i \rangle}, \tag{6.16}$$

again for $|\Delta_9| \leq 2\epsilon_0$. Paige's analysis assumes that $\sqrt{\langle \underline{\mathbf{w}_i}, \overline{\mathbf{w}_i} \rangle}$ is computed up to multiplicative accuracy $(1 + \epsilon_{\text{mach}}(n+2)/2)$ (see [Pai76] equation (12)), so as long as $2\epsilon_0 \leq \epsilon_{\text{mach}}(n+2)/2$, then there is some choice of acceptable roundoff error such that, as desired.

$$\beta'_{i+1} = \beta_{i+1}, \tag{6.17}$$

This establishes the second condition of (6.4).

Line 10. By (6.17), the if condition in Line 9 evaluates to true in Algorithm 6 if and only if it evaluates to true in Algorithm 7, so we can move onto Line 13.

Line 13. We have immediately from (6.10) and (6.17) that:

$$\overline{\mathbf{q}_{i+1}} = \mathbf{M}^{-1/2}\mathbf{q}_{i+1}. \tag{6.18}$$

This proves our third condition in (6.4) for $i+1$. Additionally, from (6.15) and (6.17), we have that:

$$\|\underline{\mathbf{q}_i} - \mathbf{M}^{1/2}\mathbf{q}_i\|_{\mathbf{M}^{-1}} = \frac{1}{\beta_i}\|\underline{\mathbf{w}_i} - \mathbf{M}^{1/2}\mathbf{w}_i\|_{\mathbf{M}^{-1}} = \frac{1}{\beta_i}\|\mathbf{M}^{-1/2}\underline{\mathbf{w}_i} - \mathbf{w}_i\|_{\mathbf{M}^{-1}} \leq \frac{\epsilon_0}{\beta_i}\|\mathbf{M}^{-1/2}\underline{\mathbf{w}_i}\|,$$

As before, we have that $\|\mathbf{M}^{-1/2}\underline{\mathbf{w}_i}\| \leq 2\|\mathbf{w}_i\|$ and, from [Pai76] equation 12, as long as $\epsilon_{\text{mach}} \leq \frac{1}{n+2}$, $\beta_{i+1} \geq \frac{1}{2}\|\mathbf{w}_i\|$. So, we conclude that

$$\|\underline{\mathbf{q}_i} - \mathbf{M}^{1/2}\mathbf{q}_i\|_{\mathbf{M}^{-1}} \leq 4\epsilon_0$$

which implies our fourth condition in (6.4) as long as $4\epsilon_0 \leq \epsilon_{\text{mach}}$.

Completing the Proof. With (6.4) in place, we are ready to prove our main result, Theorem 6.1. We have established that if Algorithm 6 is run with input $\mathbf{b}$ and function SolveM satisfying (6.1) with $\epsilon_0 \leq \epsilon_{\text{mach}}/4$, then it generates a tridiagonal matrix $\mathbf{T}'$ that is identical to the output of Algorithm 7 run with input $\mathbf{b}'$ and a particular set of round-off errors that satisfy the condition of Fact 6.2. Moreover, the matrix $\overline{\mathbf{Q}}$ generated by Algorithm 6 is identical to $\mathbf{M}^{-1/2}\mathbf{Q}$, where $\mathbf{Q}$ is the matrix generated Algorithm 7. Since we have also establishes that $z = z'$, it follows that Line 16 of each algorithm returns and identical output. We conclude from Fact 6.2 that, as long as $\epsilon_{\text{mach}} = \text{poly}(\epsilon/n\kappa_{\mathbf{M}})$, for $t = O\left(\sqrt{\kappa_{\mathbf{M}}}\log(\kappa_{\mathbf{M}}/\epsilon)\right)$, the vector $\mathbf{x}'_t$ returned by Algorithm 6 satisfies:

$$\|\mathbf{x}'_t - \mathbf{A}^{-1}\mathbf{b}'\|_{\mathbf{A}} \leq \epsilon\|\mathbf{A}^{-1}\mathbf{b}'\|_{\mathbf{A}}. \tag{6.19}$$

Recalling that $\mathbf{b}' = \mathbf{b} + \mathbf{M}\Delta_0$ for a vector Δ_0 with $\|\mathbf{M}^{1/2}\Delta_0\| \leq \epsilon_0\|\mathbf{M}^{-1/2}\mathbf{b}\|$, we further have that:

$$\begin{aligned}
\|\mathbf{A}^{-1}\mathbf{b} - \mathbf{A}^{-1}\mathbf{b}'\|_{\mathbf{A}} = \|\mathbf{A}^{-1/2}\mathbf{b} - \mathbf{A}^{-1/2}\mathbf{b}'\| &= \|\mathbf{A}^{-1/2}\mathbf{M}\Delta_0\| \\
&\leq \epsilon_0|\mathbf{A}^{-1/2}\mathbf{M}^{1/2}\|\|\mathbf{M}^{-1/2}\mathbf{b}\| \\
&\leq \epsilon_0|\mathbf{A}^{-1/2}\mathbf{M}^{1/2}\|\|\mathbf{M}^{-1/2}\mathbf{A}^{1/2}\|\|\mathbf{A}^{-1/2}\mathbf{b}\| \\
&= \epsilon_0\sqrt{\kappa_{\mathbf{M}}}\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}
\end{aligned}$$

Combining with (6.19) via triangle inequality, we have that, as long as $\epsilon_0 \leq \frac{\epsilon}{\sqrt{\kappa_{\mathbf{M}}}}$,

$$\|\mathbf{x}'_t - \mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}} \leq 3\epsilon\|\mathbf{A}^{-1}\mathbf{b}\|_{\mathbf{A}}.$$

Adjusting the constant on ϵ proves Theorem 6.1. ∎

7 Lower Bound for Linear Systems with k Large Singular Values

In this section, we give a lower bound for the time complexity of solving linear systems with k large singular values.

THEOREM 7.1. *Assuming that the time complexity of solving an arbitrary $n \times n$ linear system to precision $\epsilon = 1/\text{poly}(n)$ is $\Omega(n^\omega)$, the time complexity of solving an $n \times n$ linear system $\mathbf{Ax} = \mathbf{b}$ such that $\sigma_{k+1}(\mathbf{A})/\sigma_{\min}(\mathbf{A}) = O(1)$ is at least $\Omega(n^2 + k^\omega)$.*

Proof. The proof will be divided into two parts. First, we will give an information theoretic lower bound showing that to solve a linear system with k large singular values, one must read at least $\Omega(n^2)$ entries of the matrix $\mathbf{A}$. To do this, consider the following class of $n \times n$ matrices parameterized by an index pair (i, j), where $\mathbf{e}_i$ denotes the ith n-dimensional standard basis vector:

$$\mathcal{A} = \left\{\mathbf{I}_n + \mathbf{e}_i\mathbf{e}_j^\top - \mathbf{e}_j\mathbf{e}_i^\top \quad : \quad i \neq j\right\}.$$

Matrix $\mathbf{A} \in \mathcal{A}$ is a matrix with identity on the diagonal and then two off-diagonal entries, 1 and -1, symmetrically across from each other, and then zeros everywhere else. We then choose an all-ones vector $\mathbf{b} = [1, ..., 1]^\top \in \mathbb{R}^n$ and consider such a linear system:

$$\begin{bmatrix} 1 & & & & & & \\ & 1 & & & 1 & & \\ & & 1 & & & & \\ & & & \ddots & & & \\ & -1 & & & 1 & & \\ & & & & & 1 & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

Note that this linear system can be written more concisely as satisfying the following equations:

$$\begin{aligned} \text{for } l \neq i,j, \quad x_l &= 1, \\ x_i + x_j &= 1, \\ -x_i + x_j &= 1, \end{aligned}$$

which implies that the solution vector $\mathbf{x}$ is an all-ones vector except for $x_i = 0$. To find the singular values associated with the matrix $\mathbf{A}$, observe that:

$$\mathbf{A}^\top \mathbf{A} = (\mathbf{I}_n + \mathbf{e}_i\mathbf{e}_j^\top - \mathbf{e}_j\mathbf{e}_i^\top)^\top(\mathbf{I}_n + \mathbf{e}_i\mathbf{e}_j^\top - \mathbf{e}_j\mathbf{e}_i^\top) = \mathbf{I}_n + \mathbf{e}_i\mathbf{e}_i^\top + \mathbf{e}_j\mathbf{e}_j^\top.$$

Since $\mathbf{A}^\top\mathbf{A}$ is diagonal, its eigenvalues can be found immediately, which gives us the singular values of $\mathbf{A}$ (in decreasing order): $\sigma_1 = \sqrt{2}$, $\sigma_2 = \sqrt{2}$ and $\sigma_3 = ... = \sigma_n = 1$. Thus, $\mathbf{A}$ is well-conditioned and it satisfies the k large singular values property for any k. Suppose that the matrix $\mathbf{A}$ is given to us in an $n \times n$ two-dimensional array. How long does it take to find either of the non-zero off-diagonal entries of $\mathbf{A}$? Naturally, this will require in the worst-case reading $\Omega(n^2)$ entries, since we have no information about the index pair $1 \leq i < j \leq n$. On the other hand, if we were able to solve this linear system in time $o(n^2)$, obtaining (an approximation of) vector $\mathbf{x}$, then we could find i, j pair in an additional $O(n)$ time by finding the entry of $\mathbf{x}$ that is not equal 1. This shows, by contradiction, that the time complexity of solving such a linear system must be $\Omega(n^2)$.

We note that, while our construction gives a sparse matrix, which may be stored more efficiently in a column-row-value format, this does not circumvent our lower bound, since we can also consider the above matrix $\mathbf{A}$ that has been distorted entry-wise with, say, random $\pm 1/\text{poly}(n)$ values, so that it is no longer sparse, but this distortion will not meaningfully affect its condition number or the linear system solution. Thus, the above lower bound applies also for such dense matrices.

In the second part of the proof, we observe that to solve an $n \times n$ linear system with k large singular values, one must (effectively) solve an arbitrary $k \times k$ linear system. Namely, consider an arbitrary $k \times k$ rank k matrix $\mathbf{M}$ and a vector $\mathbf{c} \in \mathbb{R}^k$, and consider the following $n \times n$ linear system that can be written in a block-diagonal form:

$$\begin{bmatrix} \mathbf{M} & \\ & \sigma_{\min}(\mathbf{M}) \cdot \mathbf{I}_{n-k} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{0}_{n-k} \end{bmatrix},$$

where we are solving for the vectors $\mathbf{x}$ and $\mathbf{y}$. Note that the singular values of this system are simply the singular values of $\mathbf{M}$, with its smallest singular value additionally repeated $n - k$ times. Thus, $\sigma_{k+1}/\sigma_{\min} = 1$ for this system, and it satisfies the k large singular values property. Also, note that the solution of this system is $\mathbf{x} = \mathbf{M}^{-1}\mathbf{c}$ and $\mathbf{y} = \mathbf{0}$. Now, if we can solve this system in time $o(k^\omega)$, then, by simply reading off the first k entries of the solution vector, we obtain the solution to the $k \times k$ system $\mathbf{Mx} = \mathbf{c}$. This concludes the proof of the lower bound. □

Acknowledgements

MD would like to acknowledge NSF CAREER for partial support. CM was partially supported by NSF Award (2045590).

References

[AC09] Nir Ailon and Bernard Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on computing*, 39(1):302–322, 2009.

[ACG+23] Noah Amsel, Tyler Chen, Anne Greenbaum, Cameron Musco, and Christopher Musco. Near-optimal approximation of matrix functions by the lanczos method. *arXiv preprint arXiv:2303.03358*, 2023.

[ACW17] Haim Avron, Kenneth L Clarkson, and David P Woodruff. Faster kernel ridge regression using sketching and preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1116–1138, 2017.

[AL86] Owe Axelsson and Gunhild Lindskog. On the rate of convergence of the preconditioned conjugate gradient method. *Numerische Mathematik*, 48:499–524, 1986.

[AM15] Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[BKM22] Vladimir Braverman, Aditya Krishnan, and Christopher Musco. Sublinear time spectral density estimation. In *54th Annual Symposium on Theory of Computing (STOC)*, 2022.

[BT24] Ainesh Bakshi and Ewin Tang. An improved classical singular value transform for quantum machine learning. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2024.

[BV04] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[CCKW22] Nadiia Chepurko, Kenneth L Clarkson, Praneeth Kacham, and David P Woodruff. Near-optimal algorithms for linear algebra in the current matrix multiplication time. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3043–3068. SIAM, 2022.

[CD21] Xue Chen and Michal Derezinski. Query complexity of least absolute deviation regression via robust uniform convergence. In *Conference on Learning Theory*, pages 1144–1179. PMLR, 2021.

[CDDR24] 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.

[CEM+15] 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.

[CKK+18] Michael B Cohen, Jonathan Kelner, Rasmus Kyng, John Peebles, Richard Peng, Anup B Rao, and Aaron Sidford. Solving directed laplacian systems in nearly-linear time through sparse lu factorizations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 898–909. IEEE, 2018.

[CLS21] Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM (JACM)*, 68(1):1–39, 2021.

[CMM17] 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.

[CNW16] Michael B Cohen, Jelani Nelson, and David P. Woodruff. Optimal Approximate Matrix Product in Terms of Stable Rank. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55, pages 11:1–11:14, 2016.

[Coh16] Michael B Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 278–287. SIAM, 2016.

[CP15] Michael B Cohen and Richard Peng. Lp row sampling by lewis weights. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 183–192, 2015.

[CW87] 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.

[CW13] Kenneth L Clarkson and David P Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of Computing*, pages 81–90, 2013.

[DEF+23] Mateo Díaz, Ethan N. Epperly, Zachary Frangella, Joel A. Tropp, and Robert J. Webber. Robust, randomized preconditioning for kernel ridge regression. *arXiv preprint arXiv:2304.12465*, 2023.

[DLDM21] Michal Derezinski, Zhenyu Liao, Edgar Dobriban, and Michael Mahoney. Sparse sketches with small inversion bias. In *Conference on Learning Theory*, pages 1467–1510. PMLR, 2021.

[DLNR24] Michał Dereziński, Daniel LeJeune, Deanna Needell, and Elizaveta Rebrova. Fine-grained analysis and faster algorithms for iteratively solving linear systems. *arXiv preprint arXiv:2405.05818*, 2024.

[DM24] 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.

[DR24] 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.

[DW18] Edgar Dobriban and Stefan Wager. High-dimensional asymptotics of prediction: Ridge regression and classification. *The Annals of Statistics*, 46(1):247–279, 2018.

[DY24] Michał Dereziński and Jiaming Yang. Solving dense linear systems faster than via preconditioning. In *56th Annual ACM Symposium on Theory of Computing (STOC)*, 2024.

[EAM14] Ahmed El Alaoui and Michael W Mahoney. Fast randomized kernel methods with statistical guarantees. *stat*, 1050:2, 2014.

[Epp24] Ethan N. Epperly. Fast and forward stable randomized algorithms for linear least-squares problems. *arXiv preprint arXiv:2311.04362*, 2024.

[FGKS15] Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, pages 2540–2548, 2015.

[FMMS16] Roy Frostig, Cameron Musco, Christopher Musco, and Aaron Sidford. Principal component projection without principal component analysis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pages 2349–2357, 2016.

[FTU23] Zachary Frangella, Joel A Tropp, and Madeleine Udell. Randomized Nyström preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 44(2):718–752, 2023.

[GM16] Alex Gittens and Michael W. Mahoney. Revisiting the Nyström method for improved large-scale machine learning. *J. Mach. Learn. Res.*, 17(1):3977–4041, 2016.

[GOSS16] Alon Gonen, Francesco Orabona, and Shai Shalev-Shwartz. Solving ridge regression using sketched preconditioned SVRG. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1397–1405, 2016.

[Gre89] Anne Greenbaum. Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences. *Linear Algebra and its Applications*, 113:7 – 63, 1989.

[GY99] Gene H Golub and Qiang Ye. Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM Journal on Scientific Computing*, 21(4):1305–1320, 1999.

[HMT11] 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.

[HS52] Magnus R Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

[Hut90] Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990.

[JJM23] Ruichen Jiang, Qiujiang Jin, and Aryan Mokhtari. Online learning guided curvature approximation: A quasi-newton method with global non-asymptotic superlinear convergence. *arXiv preprint arXiv:2302.08580*, 2023.

[JKL+20] Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. A faster interior point method for semidefinite programming. In *2020 IEEE 61st annual symposium on foundations of computer science (FOCS)*, pages 910–918. IEEE, 2020.

[JS19] Yujia Jin and Aaron Sidford. Principal component projection and regression in nearly linear time through asymmetric svrg. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[JZ13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

[KKM79] Thomas Kailath, Sun-Yuan Kung, and Martin Morf. Displacement ranks of matrices and linear equations. *Journal of Mathematical Analysis and Applications*, 68(2):395–407, 1979.

[KMP12] Ioannis Koutis, Gary L Miller, and Richard Peng. A fast solver for a class of linear systems. *Communications of the ACM*, 55(10):99–107, 2012.

[KMS+22] Jonathan Kelner, Annie Marsden, Vatsal Sharan, Aaron Sidford, Gregory Valiant, and Honglin Yuan. Big-step-little-step: Efficient gradient methods for objectives with multiple scales. In *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178, pages 2431–2540, 2022.

[KN14] Daniel M Kane and Jelani Nelson. Sparser johnson-lindenstrauss transforms. *Journal of the ACM (JACM)*, 61(1):1–23, 2014.

[KS16] Rasmus Kyng and Sushant Sachdeva. Approximate gaussian elimination for laplacians - fast, sparse, and simple. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 573–582. IEEE Computer Society, 2016.

[Lev44] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

[LG12] Francois Le Gall. Faster algorithms for rectangular matrix multiplication. In *IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 514–523, 2012.

[LGW+21] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, 2021.

[LL10] Dennis Leventhal and Adrian S Lewis. Randomized methods for linear constraints: convergence rates and conditioning. *Mathematics of Operations Research*, 35(3):641–654, 2010.

[LS13] 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.

[LW11] Po-Ling Loh and Martin J Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *Advances in neural information processing systems*, 24, 2011.

[Mar63] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[MB17] Siyuan Ma and Mikhail Belkin. Diving into the shallows: a computational perspective on large-scale shallow learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[MCRR20] Giacomo Meanti, Luigi Carratino, Lorenzo Rosasco, and Alessandro Rudi. Kernel methods through the roof:

Handling billions of points efficiently. In *Advances in Neural Information Processing Systems*, volume 33, pages 14410–14422, 2020.

[MM13] Xiangrui Meng and Michael W Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 91–100, 2013.

[MM15] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. *Advances in neural information processing systems*, 28, 2015.

[MM17] Cameron Musco and Christopher Musco. Recursive sampling for the nystrom method. *Advances in neural information processing systems*, 30, 2017.

[MMMW21] Raphael A. Meyer, Cameron Musco, Christopher Musco, and David P. Woodruff. Hutch++: Optimal stochastic trace estimation. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 142–155, 2021.

[MMS18] 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, 2018.

[MNS+18] 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, 2018.

[MS12] Renato DC Monteiro and Benar F Svaiter. Iteration-complexity of a newton proximal extragradient method for monotone variational inequalities and inclusion problems. *SIAM Journal on Optimization*, 22(3):914–935, 2012.

[MW17] Cameron Musco and David P. Woodruff. Sublinear time low-rank approximation of positive semidefinite matrices. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 672–683, 2017.

[NN13] 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.

[Nys30] E. J. Nyström. Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben. *Acta Math.*, 54:185–204, 1930.

[OSV12] Lorenzo Orecchia, Sushant Sachdeva, and Nisheeth K Vishnoi. Approximating the exponential, the lanczos method and an o (m)-time spectral algorithm for balanced separator. In *44th Annual Symposium on Theory of Computing (STOC)*, 2012.

[Pai71] Christopher C. Paige. *The computation of eigenvalues and eigenvectors of very large sparse matrices*. PhD thesis, University of London, 1971.

[Pai76] Christopher C. Paige. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *IMA Journal of Applied Mathematics*, 18(3):341–349, 1976.

[Pan84] Victor Pan. *How to multiply matrices faster*. Springer-Verlag, 1984.

[PV21] Richard Peng and Santosh Vempala. Solving sparse linear systems faster than matrix multiplication. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2021.

[RCR17] Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. Falkon: An optimal large scale kernel method. *Advances in neural information processing systems*, 30, 2017.

[RSB12] Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, pages 2663–2671, 2012.

[RT08] 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.

[Sar06] 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.

[She94] Jonathan R Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Carnegie Mellon University, USA, 1994.

[SSZ14] Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 64–72, 2014.

[ST14a] Daniel A Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

[ST14b] Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

[Str69] Volker Strassen. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354–356, 1969.

[SV09] Thomas Strohmer and Roman Vershynin. A randomized kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278, 2009.

[SW09] Daniel A Spielman and Jaeoh Woo. A note on preconditioning by low-stretch spanning trees. *arXiv preprint*

arXiv:0903.2816, 2009.

[Tro11] Joel A Tropp. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126, 2011.

[Wil12] 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.

[WS01] Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. 2001.

[WW10] Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 645–654, 2010.

[WXXZ23] Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. *arXiv preprint arXiv:2307.07970*, 2023.

[XXG12] Jianlin Xia, Yuanzhe Xi, and Ming Gu. A superfast structured solver for toeplitz linear systems via randomized sampling. *SIAM Journal on Matrix Analysis and Applications*, 33(3):837–858, 2012.

[ZDW13] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression. In *Conference on learning theory*, pages 592–617. PMLR, 2013.